V3.0

# مسائل برمجة محلولة

## خطوة على طريق البرمجة


## كلية الهندسة المعلوماتية

## إعداد محمد وسيم أبوزينة

## wazzz


- تم تطبيق معظم خوارزميات البرامج على الحاسب لذلك تاكد عزيزي القارىء من صحتها فستجد اشارة Not compiled بجانب اي المسالة لم يتم تجريبها وفي حال أي اخطاء يمكن مراسلتي عبر البريد الإلكتروني wase39@gmail.com

الفصل الأول :
وظائف عملي برمجة 1 سابقة محلولة.

الفصل الثاني
مسائل خارجية تساعد لفهم افكار النظري

الفصل الثالث
مسائل التعامل مع المحارف Chars

الفصل الرابع
مسائل المصفوفات Arrays

الفصل الخامس
مسائل العودية Recursion

الفصل السادس
مسائل الأعداد متقدم Advanced Numeric

الفصل السابع
مسائل مقرر البرمجة1 العامة

الفصل الثامن
مسائل دورات برمجة 1 في كلية الهندسة المعلوماتية

## ملاحظة هامة

لن تحقق الفائدة المرجوة من الكتاب إن لم تقم بحل المسائل لوحدك واستعنت بالحل الجاهز

**الفصل الأول: وظائف عملي برمجة 1**

**المسألة 1:**

**اكتب برنامج بلغة** Pascal.T **لقراءة عددين صحيحين وطباعة نواتج عملية الضرب والقسمة والجمع والطرح لهذين العددين**

```
Program   Calc ;
Var  x,y:integer;

Begin
    Writeln('Enter Two number ' );
    Readln(x,y);
    Writeln('The Result of  ',x,'   +',y, '     =',x+y);
    Writeln('The Result of  ',x,'   *',y, '     =',x*y);
    Writeln('The Result of  ',x,'   -',y, '     =',x-y);
    Writeln('The Result of  ',x,'  /',y, '     =',x/y);
End.
```

**المسألة 2:**

**اكتب برنامج بلغة** pascal.T **يقوم بقراءة عدد صحيح** X **وعدد صحيح موجب** n **ثم يقوم بحساب** Xn

```
Program   Power_To_ppl ;
Var  x,n,i,Res:integer;

Begin
Res =:1;
Writeln('Enter Two numbers X, and n ');
Readln(x,n)
For i=1 to n do
   Res =Res *X;

Writeln('The Result is ',Res);
End.
```

**المسألة 3 :**

**اكتب برنامج لقلب عدد صحيح مدخل مثال العدد المدخل1234 يصبح 4321**

```
Program   Flip_Flop ;
Var  n,m,num:integer;
Begin
Writeln(‘Enter The Number ‘);
Readln(n);
while (n div 10 <>0 )do
begin
      m       =:n mod 10;
      num  =:m+num*10;
      n        =:n div 10;
end;
Writeln(‘The Number After Flipping ‘,n);
end.
```

**المسألة 4 :**

**اكتب برنامج يقوم بحساب قيمة تقريبية لـ ex بحيث**

$$e^x = \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \ldots \ldots \ldots ..$$

**مع العلم انه يتوقف الحساب عندما تكون قيمة الحد الأخير الذي نصل إليه اصفر أو تساوي** $10^{-5}$

```
Program   Damn_The_e  ;
Var  x,n,i,counter  :integer;
       Power,fact ,res:real
Begin
fact =:1;      counter=2;         res =0;
Writeln(‘Enter Two numbers X ‘);
Readln(x);
Power=x;
While (power div fact < 10 ^(-5)) do
  begin
        For i=2 to counter do
             begin
                     power =power *X;
                     fact = fact *i;
              end;
        Counter = counter +1;
        Power =x;                     fact= 1;
         Res= Res + (power div fact);
   End;
Writeln(‘The Result is ‘,Res);
End.
```

**المسألة 5 :**

**1-أكتب خوارزمية تقوم بقراءة عددين A,B و طباعة الأعداد المحصورة بينهما بطريقة تنازلية دون معرفة أي العددين A,B هو الأكبر .**

**2-أكتب خوارزمية تقوم بقراءة عدد N و طباعة مجموع الأعداد الزوجية المحصورة بين n و الصفر .**

**3-أكتب خوارزمية تقوم بقراءة رقم m ما و طباعة عدد خاناته .**

```
program  Numberz;
var
                  Digit_counter,m, i,a,b,min,max,sum,first,last:integer;

begin
Readln(a,b);
If (a>b)then
     Max =a;  min =b
Else
     max=b;   min =a
first=min ; last =max
while (first <=max ) do
   begin
      last = last -1
      fisrt = first +1
      writeln(last)
   end;
end.
2:
Readln(n)
Sum=0;
For i=0 to n do
   If i mod 2 =0 then
        Sum= sum+i;
3:
Readln(m)
Digit_counter=0;
While (m <>0) do
      begin
           Digit_counter =  Digit_counter+1;
          M = m div 10;
      End;
 writeln(m);
End.
```

**المسألة 6 :**

1- **وضع أحد العملاء مبلغ** sum **في البنك لعدد** year **من السنوات ..**
**ما هي قيمة هذا المبلغ بعد هذه السنوات علماً أن العميل يتقاضى على المبلغ فائدة [ مركبة ] سنوية** p%

2-**وضع أحد العملاء مبلغ** sum **في البنك فإذا علمت أن العميل يتقاضى على المبلغ فائدة [ مركبة ] سنوية** p%
**بعد كم سنة يتضاعف المبلغ ..؟؟؟**

```
program Banks;
var i,num,year,value:integer;
p :float ;

begin
1:
Readln(year,num,p)
For i=1 to year do
   Num = num + Num *p;
2:
Y=0;                  Value =2*num;
While (num<= value )do
        begin
              num= num +num *p;
              Y=Y+1;
         End;
Writeln(y);

end;
```

**المسألة 7**

**لدينا فاتورة مشتريات تحوي البيانات التالية :**

**عدد المواد** N

**عدد الوحدات من كل مادة** q

**سعر الوحدة** unit price

**يقوم التاجر بحسم جزء من قيمة الفاتورة إن كانت هذه القيمة مرتفعة كما يلي :**

**حسم** 10 **٪ إذا كانت قيمة الفاتورة أكبر من** 25000

**حسم** 15 **٪ إذا كانت قيمة الفاتورة أكبر من** 50000

**حسم** 20 **٪ إذا كانت قيمة و الفاتورة أكبر من** 100000

**و المطلوب:**

**كتابة برنامج يقوم بقراءة البيانات السابقة ثم حساب و طباعة:**

**السعر الإجمالي** total price **[ و هو قيمة الفاتورة قبل الحسم ].**

**السعر الصافي** net price **[ قيمة الفاتورة بعد الحسم ].**

```
program Bills ;
var
          N,q,Unit_price,Total,net_price :integer;
q=0;   Unit_price=0 ;   Total =0;     net_price=0  ;
Readln(n);
For i =1 to n do
    begin
          Writeln('Enter the price for the product num ',i);
          Readln(Unit_price);
          Writeln('Enter the Quantity  for the product num ',i);
          Readln(q);
          Total = total +q* unit_price;
     End;
If (tp> 100000)
             Netprice = (total * 20)/100;
Else If (tp> 50000 )
              Netprice = (total * 15)/100;
       Else If (tp> 25000)
              Netprice = (total * 10)/100;

writeln(Netprice,'    ', 'total ');
end.
```

**المسالة 8 :**

**اكتب برنامج لقراءة نص محرف محرف ينتهي بنقطة [.] والمطلوب :**

1- **حساب عدد المحارف الكلي**

2- **حساب عدد الأحرف وعدد الأرقام وعدد علامات التنقيط**

3- **طباعة النص مع استبدال كل حرف صغير في اول الكلمة بحرف كبير**

4- **تحويل كل عدد إلى عدد عشري حيث ان العدد العشري** =n-9 **أي المتمم إلى** 10

```
program  Reading with the stars ;
var                          ch,ch1 : char;
                              countchr: integer;       // Char counter
                              countnum: integer;      // Numbers Counter
                              counter3: integer;       // Other Char Counter
begin
writeln('Enter the text');
          countchr:=0;    countnum:=0;       counter3:=0;
while (ch <> '.') do
        begin
           ch1:=ch;
           read (ch);
         case ch of
           '0'..'9' :
                begin
                     ch:=chr(ord('9')-ord(ch) + ord('0') ) ;  //Converting Number
                     countnum:=countnum+1;
                end;
          'a'..'z':
              begin
                   countchr:= countchr+1;
                   ch:=chr(ord('A')+ord(ch)-ord('a'));  // Converting to Capital
               end;
                  if (ch1 = ' ') or (ch1= ',') or (ch1= ' "  ') then
                       counter3:=counter3+1;
         end;  // While
 write(ch);
end;

writeln('The number of the letters  =',countchr);
writeln('The number of numbers are  = ',countnum);
writeln('Number of the symbols  =  ',counter3+countnum+countchr);

end.        // End of program
```

**المسألة 9 :**

**اكتب برنامج يعطي نتيجة منطقية True أو False عن تناظر مصفوفة أحادية بعدها n بالنسبة لمنتصفها مثال**

**نقول عن مصفوفة إحادية إنها متناظرة**

**في حال بعد المصفوفة**

**فردي** N=9

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2 | 7 | 5 | 6 | 4 | 6 | 5 | 7 | 2 |

**زوجي** N=8

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 7 | 5 | 6 | 6 | 5 | 7 | 2 |

**نقول عن مصفوفة إحادية إنها غير متناظرة**

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 2 | 5 | 6 | 7 |

```
program   Semitic _Arrays;

const maxn=100;

var                          A:array [1..maxn]of integer;
                             b:boolean;
                             n:integer;
                             i:integer;
begin
writeln('Please enter the Array Length ');
readln(n);
writeln('Please enter the elements so The Array can be initialized ‘);

For i:=1 to n do
      readln(A[i]);
b:=true;
For i:=1 to n do
       if (A[i]<>A[n-i+1]) then
              b:=false;

if (b=true) then
       writeln('The matrix is  Semitic  ‘);
else if (b=false) then
       writeln('The matrix is not  Semitic  ‘);
readln;
end.
```

**المسألة 10 :**

**يتم حساب رواتب الموظفين في دائرة حكومية بالشكل التالي :**

**لكل موظف راتب اساسي يضاف إليه قيمة تعويضات يستحقها حسب نشاطه وعمله وتخصم منه قيمة حسميات حسب إجازاته وعطله اكتب برنامج بلغة باسكال ينظم جداول رواتب الموظفين يطلب فيه**

1-**قراءة الأرقام التسلسلية للموظفين وتخزينها في جدول ارقام الموظفين**

| ارقام الموظفين |
|---|

2-**قراءة تفصيل راتبه حيث يتم التخزين في 3 جداول كالتالي:**

**الراتب الأساسي – التعويضات – الحسميات**

| الراتب الأساسي |
|---|

| التعويضات |
|---|

| الحسميات |
|---|

3-**حساب إجمالي الراتب للموظفين وتخزينه في جدول**

**حيث اجمالي الراتب = الراتب الأساسي + التعويضات – الحسميات**

| إجمالي الراتب |
|---|

4-**إظهار اكبر قيمة للراتب الأساسي**

**إظهار رقم الموظيف الذي يتقاضى اعلى قيمة تعويضات**

**إظهار رقم الوظف الذي يخصم من راتبه اقل قيمة حسميات**

**حساب وسطي الراتب الإجمالي**

**وتخزين هذه القيم في جدول النتائج التالي:**

**النتائج**

**اكبر قيمة للراتب الأساسي – رقم الموظف الذي له اعلى تعويضات – رقم الموظف الذي له اقل حسميات – وسطي الراتب الإجمالي**

| اكبر قيمة للراتب الأساسي | رقم الموظف الذي له اعلى تعويضات | رقم الموظف الذي له اقل حسميات | وسطي الراتب الإجمالي |
|---|---|---|---|
| | | | |

5-**إظهار النتائج السابقة بالشكل**

**رقم الموظف – الراتب الأساسي – التعويضات – الحسميات – إجمالي الراتب**

**النتائج**

| رقم الموظف | – الراتب الأساسي | التعويضات | – الحسميات | إجمالي الراتب |
|---|---|---|---|---|

**النتائج**

| النتائج |
|---|

```
Program employee ;
Const    m=100;
Type Matrix =array[1..m] of integer;
Var
          Num,basic,comp,dis,total:matrix;
          I,j,max1,max2,min,sum,x,y,z:integer;
          Avg:real

Begin  {main }
  Writeln('Please Enter the whole numbers of  employees');
   Readln(z);
    Writeln('Please Enter the number of the employee');
For j =1 to z do
begin
       Readln(Num[j]);
End;

Writeln('Please Enter the basic Salary for each  employee');
For j=1 to z do
begin
        Readln(Basic[j]);
End;

Writeln('Plz Enter the compensations for each  employee');
For j=1 to z do
begin
       Readln(Comp[j])
End;

Writeln('Please Enter the Discounts for each  employee');
For j=1 to z do
begin
       Readln(Dis[j]);
End;

For j=1 to z do
   Total[j]=Basic[j]+Comp[j]-Dis[j]

Writeln('employee No    Basic Salary    Bounces     Discounts    Total   ');
Writeln;
   For j=1 to z do
     Begin
         Write(Num[j]:5,'   ',Basic[j]:5,'   ',Comp[j]:5,'    ',Dis[j]:5,'   ',Total[j]:5);
         Writeln;
     End;
```

```
Max1=Basic[1];

Mx2=Comp[1];

X=num[1];

Y =num[1];

Min=dis[1];

Sum=0;

For j=1 to z do
Begin
If Basic[j]>max1 then
    Max1=Basic[j];

If comp[j]>max2 then
    begin
          Max2= comp[j];
           X=num[j];
    End;

If dis[j]<min then
   Begin
           Min= dis[j];
           Y=num[j];
   End;
Sum= sum + total[j];
End;
Avg= sum/z;
Write(max1,x,y,avg);

End.
```

**المسألة** 11:

**لدينا مصفوفة مربعة بعدها** n.n **يتم إدخال البعد** n **من قبل المستثمر والمطلوب**

1-**حساب مجموع عناصر محيط هذه المصفوفة**

2- **حساب مجموع عناصر الأسطر** m **الأخيرة من هذه المصفوفة**

3-**حساب مجموع عناصر المثلث العلوي في المصفوفة المربعة ما عدا القطر الرئيسي**

4-**فرز عناصر القطر الثانوي لمصفوفة مربعة تصاعدياً**

```
program Matrix ;
const z= 100;
var                                   A: array [1..z,1..z]of integer;
                                      B: array [1..z,1..z]of integer;
                                     t, v,m,sumv, sum1,sum2, n,i,j:integer;
                  f,k,l, sumq1, sumq2, summ, sumrec, sum, sum11,sum22:integer;
                                                g:boolean;
begin
writeln('Enter the length of the array');
readln(n);

{reading the Array}
for i:=1 to n do
  for j:=1 to n do
     readln(A[i,j]);

//  جمع عناصر السطر الأخير والسطر الأول والعمود الأول والأخير
sum1:=0;
for i:=1 to n do
   sum1:=(sum1+A[i,1]);

 sum11:=0;

for i:=1 to n do
   sum11:=sum11+A[i,n];

sum2:=0;
for j:=1 to n do
  sum2:=(sum2+A[1,j]);

 sum22:=0;
for j:=1 to n do
  sum22:=sum22+A[n,j];

//مجموع المحيط هو مجموع الأربعة اسطر واعمدة
sum:=sum1+sum11+sum22+sum2-A[1,1]-A[1,n]-A[n,n]-A[n,1];
writeln('the sum of the circumfence is',sum);

// طباعة المصفوفة بشكل ثنائي

for i:=1 to n do

begin
     write(' ');
    for j:=1 to n do
    write(A[i,j],' ');
    writeln(' ');
```

```
end;

writeln('Enter m number of lines to show u the sum');
readln(m);
if (m>n) or (m<0) then

     begin
          writeln('please try again');
          g:=false;
          readln(m);
          while (m>n) do
          readln(m)
     end
else v:=n-m;
    for j:=v+1 to n do
       for i:=1 to n do
             sumv:=sumv+A[j,i];
writeln('the sum of the last ',m,' lines is ',sumv);

//الطلب ال3

f:=2; sumrec:=0;
for i:=1 to n do
   begin
     for j:=f to n do
       sumrec:=sumrec+A[i,j];
     f:=f+1;
   end;
writeln('the sum for the rectangle is ',sumrec);

for f := 1 to n-1 do
   begin
     j:= n;
     for i := 1 to n-f do
        begin
         if a[i,j] < a[i+1,j-1] then
             begin
                t := a[i,j];
                a[i,j] := a[i+1,j-1];
                a[i+1,j-1] := t;
             end;
     j:= j-1
         end;
    end;
for i := 1 to n do
     begin
        writeln(' ');
        for j := 1 to n do
        write(a[i,j] , ' ');
        writeln(' ');
     end;
readln;
end.

End.
```

**المسألة** 12:

1-**اكتب تابع يقوم بفحص مصفوفة مربعة بعدها** n **فيما إذا كانت متناظرة أم لا**

2-**اكتب إجرائية تقوم بإيجاد القيمة العظمى لعناصر مصفوفة متناظرة**

3-**اكتب تابع يقوم بإيجاد القيمة العظمى لعناصر مصفوفة غير متناظرة**

4-**اكتب برنامجاً يستدعي التوابع والإجرائيات السابقة لإيجاد القيمة العظمى لعناصر مصفوفة ما**

```
program  Matrix_part2;
type mat=array[1..10,1..10]of integer;
var                                                        a:mat;
                                                   n,i,j,max:integer;
                                                      ok:boolean;


{1}

function simitric(n:integer):boolean;
           var i,j:integer;
           error:boolean;
begin
    i:=1;
   error:=false;
   while (i<=n ) and (not error) do
       begin
           j:=1;
          while (j<=n) and (not error) do
             begin
                  if a[i,j]<>a[j,i] then
                      error:=true
                 else
                       j:=j+1;
             end;
         i:=i+1;
        end;
simitric:=not error;
end;

{end fo function simitric}
{2}

procedure max1(n:integer;var max:integer);
                         var i,j:integer;
begin
max:=a[1,1];
   for i:=1 to n do
      for j:=1 to n do
          if a[i,j]>max then
              max:=a[i,j];
end;

{end for procedure max1 }
```

```
{3}

function max2(n:integer):integer;
                      var i,j,max:integer;
begin
max:=a[1,1];
for i:= 1 to n do
     for j:=1 to n do
          if a[i,j]>max then
                max:=a[i,j];
         max2:=max;
end;

{ end fo function max2}



{4}
begin {main  program}
writeln('please inter long the matrix');
readln(n);
for i:=1 to n do
   for j:=1 to n do
       readln(a[i,j]);


ok:=simitric(n);
 if ok=true then
       writeln('the matrix is simitric')
else
        writeln('the matrix is not simitric');

max1(n,max);
writeln('the max value in the simitric matrix by procedure =',max);
writeln('the max value in the simitric by function =', max2(n));
writeln;
for i:=1 to n do
     begin
        for j:=1 to n do
             write(a[i,j],' ');
        writeln;
     end;
readln;
End.
```

**المسألة 13 :**

**1-اكتب اجرائية تاخذ كوسيط سلسلة محارف تمثل تعبير رياضي وتعيد نتيجة منطقية تدل على صحة هذا التعبير أو عدمه**

**علماً ان سلسلة المحارف تقبل فقط محارف رقمية 0-9 وتقبل العمليات *و-و+**

**2-اكتب تابع ياخذ كوسيط سلسلة محارف تمثل تعبير رياضي ويعيد قيمة هذا التعبير**

**علماً ان سلسلة المحارف تقبل فقط محارف رقمية 0-9 وتقبل العمليات *و+و-**

**3-اكتب برنامج يقرا سلسلة محارف ويستدعي الإجرائية الأولى للتأكد إذا كانت سلسلة المحارف هذه تمثل تعبير رياضي أو لا وفي حال كانت تعبير عن تعبير رياضي فيطلب استدعاء التابع الثاني لإيجاد قيمة هذا التعبير**

**الحل:**

**بصراحة لم استطع ان احل الوظيفة التالية بأفكار البرمجة 1 لذا استعنت بمسالة محلولة من قبل صديق لي وقد اعتمد افكار البرمجة 2 واستعمال والمؤشرات هذا الحل يتضمن مؤشرات وبعض افكار البرمجة 2 لذا لا داعي لفهم الكود حالياً لكن لكي يبقى للذكرة كما ان افكاره مهمة جداً للبرمجة 2**

```
program MatExp;

type
 PStack=^Element;
 Element=record
  val:string;
  prev:PStack;
 end;

var
 a:string;
 x,c:char;
 elem:Element;
 n,n1,n2,b:integer;
 top,auxtop,ptop,Rtop:PStack;


         procedure Push(var ptop:PStack; elem:Element);
         var
          temp:PStack;
         begin
              new(temp);
              temp^:=elem;
              temp^.prev:=ptop;
              ptop:=temp;
         end;
```

```
procedure Clear(var ptop:PStack);
var
 temp:PStack;
begin
      while (ptop<>nil) do
        begin
           temp:=ptop;
           ptop:=ptop^.prev;
          dispose(temp);
        end;
end;

function StrToIntConvert(a:string; n:integer):integer;
var
 conc,i,j,s:integer;
begin
   conc:=0;
   j:=1;
   while (j<=n) do
       begin
         s:=(ord(a[j])-ord('0'));
        for i:=j+1 to n do
                s:=s*10;
        conc:=conc+s;
           j:=j+1;
       end;
  StrToIntConvert:=conc;
end;

function Calc(n1,n2:integer; c:char):integer;
var
 sum:integer;
begin
  case c of
   '+': sum:=n1+n2;
   '*': sum:=n1*n2;
  end;
 Calc:=sum;
end;
```

```
function IntToStrConvert(b:integer):string;
var
 t,c,i,n,d:integer;
 s:string;
begin
      t:=b;
     n:=0;
    while (t<>0) do
       begin
          t:=t div 10;
          n:=n+1;
      end;
  c:=1;
 for i:=2 to n do
    c:=c*10;
 i:=1;
 s:=' ';
 while (i<=n) do
   begin
      d:=b div c;
      s:=s+chr(ord('0')+d);
      b:=b-(d*c);
       c:=c div 10;
       i:=i+1;
    end;
 delete(s,1,1);
 IntToStrConvert:=s;
end;

function Combine(var ptemp:PStack):string;
var
 bo:boolean;
 s:string;
begin
  s:=ptemp^.val;
  bo:=true;
  while (bo) do
  begin
 if ((ptemp^.prev^.val='1') or (ptemp^.prev^.val='2') or (ptemp^.prev^.val='3') or (ptemp^.prev^.val='4')
           or (ptemp^.prev^.val='5') or (ptemp^.prev^.val='6') or (ptemp^.prev^.val='7') or
(ptemp^.prev^.val='8')      or (ptemp^.prev^.val='9')) then
    begin
        s:=s+ptemp^.prev^.val;
        ptemp:=ptemp^.prev;
    end
     else
```

```
      bo:=false;
    end;
    Combine:=s;
  end;

  procedure Transfer(var top,auxtop:PStack);
  var
   d,Ssum:string;
   temp,temp1:PStack;
   check1,check2:boolean;
   num1,num2,sum:integer;
  begin
    temp:=top;
    while (temp<>nil) do
    begin
      if (temp^.val='(') then
      begin
        temp1:=temp^.prev;
        check1:=true;
        check2:=false;
        while (temp1^.val<>')') and (check1) do
        begin
         if (temp1^.val='(') then
           check1:=false;
         if ((temp1^.val ='+') or (temp1^.val='*')) then
           check2:=true;
        temp1:=temp1^.prev;
        end;

        if (check1) and (check2) then
        begin
         temp:=temp^.prev;
         sum:=0;
         while (temp^.val<>')') do
         begin
           if ((temp^.val<>'+') and (temp^.val<>'*')) then
           begin
            d:=Combine(temp);
            num1:=StrToIntConvert(d,length(d));
            temp:=temp^.prev;
            c:=temp^.val[1];
            temp:=temp^.prev;
            d:=Combine(temp);
            num2:=StrToIntConvert(d,length(d));
            sum:=Calc(num1,num2,c);
            temp:=temp^.prev;
```

```
          end
         else
         begin
          num1:=sum;
          c:=temp^.val[1];
          temp:=temp^.prev;
          d:=Combine(temp);
          num2:=StrToIntConvert(d,length(d));
          sum:=Calc(num1,num2,c);
          temp:=temp^.prev;
         end;
        end;
        Ssum:=IntToStrConvert(sum);
        elem.val:=Ssum;
        push(auxtop,elem);
       end
       else
        push(auxtop,temp^);
     end
     else
      push(auxtop,temp^);
    temp:=temp^.prev;
   end;
  clear(top);
 end;

procedure AuxTransfer(var top,auxtop:PStack);
var
 temp:PStack;
begin
 temp:=auxtop;
 while (temp<>nil) do
 begin
  push(top,temp^);
  temp:=temp^.prev;
 end;
 clear(auxtop);
end;
```

```
Begin
 top:=nil;
 Rtop:=nil;
 auxtop:=nil;

writeln('plz insert a mathematical expression between parentheses and ending with "." ');
 read(x);

 while (x<>'.') do
 begin
  elem.val:=x;
  push(Rtop,elem);
  read(x);
 end;

  AuxTransfer(top,Rtop);

 while (top^.prev<>nil) do
 begin
  Transfer(top,auxtop);
  AuxTransfer(top,auxtop);
 end;

 writeln;
 writeln('after calculating ... the conclusion = ', top^.val);

 readln;
 readln;

End.
```

## الفصل الثاني مسائل خارجية

المسائل التالية هي مسائل متدرجة بالصعوبة وينصح بقراءة أسس قواعد اللغة باسكال ثم البدء بهذه التمارين طبعا اغلب الحلول هي خوارزمية وليس برنامج كامل وذلك لاختصار حجم الكود وحجم الصفحات

**المسألة 1:**

**اكتب برنامج يقرأ 4 أعداد ويجد الأكبر بينهم**

```
Read ( a,b,c,d )
If (a  >b ) then
   Max1 =a;
Else
   Max1=b;
If (c>d)then
   Max2=c;
Else
   Max2= d;
If (max2 >max1 )
   Max1 = Max2
Writeln( Max1)
```

**ملاحظة برمجية عند وجود شرطين if متتالين كما في مثالنا فإن الحاسب سيقوم بفحص الشرطين معاً وفي حال عدم تحقق احدهما سينتقل إلى else الموافقة لل if للشرط المختل**

**طريقة أخرى**

```
If (a>b ) And (a>c ) And (a>d) then
    Max =a
Else if (b>c) And (b>d) then
    Max =b
Else if (c>d)
    Max=c
Else max=d
Writeln(max)
```

**المسألة 2:**

**اكتب برنامج يقرأ مجموعة من الأعداد ويطبعا وإذا ادخل 10 يتوقف البرنامج عن العمل**

```
Readln(n)
While (n <> 10 ) do
   Writeln(n);
   Readln(n);
```

أو عن طريق التعليمة التالية break التي تؤمن خروج من الحلقة غير المنتهية

```
While (true)
     Readln(n)
      If (n = 10)
           Break
     Writeln(n)
```

**المسألة 3:**

**اكتب برنامج يبدل بين قيمتين مثال إذا كانت a=3 و b=2 يصبح a=2 و b=3**

الطريقة الأولى :عن طريق 3 متحولات

```
Temp = a;
a=b;
b=Temp;
```

الطريقة الثانية : عن طريق متحولين فقط

```
a =a-b
b =a+b
a =b-a
```

| | | | |
|---|---|---|---|
| A =:a-b | A =3-2=1 | Temp=a | Temp=3 |
| B =:a+b | B =1+2=3 | A=b | A=2 |
| A =:b-a | A =3-1=2 | B=temp | B = 3 |

**المسألة 4:**

**السنة الكبيسة هي سنة يكون فيها عدد أيام شهر شباط 29 يوماً بدلا من 28 يوم، تكون السنة كبيسة إذا كانت تقبل القسمة على 4 بدون باقي، باستثناء السنوات التي تأتي في نهاية كل قرن حيث يجب ان تقبل القسمة على 400 بدون باقي لتكون كبيسة وهكذا فإن السنوات 1996 و 1940 و 200 مثلا هي سنوات كبيسة في حين السنوات مثل 1995 و 1969 و 1900 و 1800 ليست كبيسة . اكتب برنامج بلغة شبه الرماز لقراءة السنة وطباعة إن كانت كبيسة ام لا**

```
Readln( year )
Isleap= false;
If (year mod 4 = 0 )then
    If (year mod 400 =0 )then
        Isleap= true
    Else
        If (year mod 100 =0 )then
            Isleap=false;
        Else isleap=true;
Else
    Isleap = false
```

السنة تقبل القسمة على 4 و 400 = كبيسة
السنة تقبل القسمة على 4 ولا تقبل القسمة على 400 وتقبل على 100 = ليست كبيسة
السنة تقبل القسمة على 4 ولا تقبل على 400 ولا 100 = كبيسة

**المسألة 5:**

**لدى عميل في بنك مبلغ قيمته A يتقاضى فائدة سنوية قيمتها p لعدد من السنوات y احسن واطبع قيمة المبلغ بعد انقضاء عدد السنوات y**

```
Read A,p,y
C=1
While (c<= y )do
    A = A+A*p
    C= C+1
Writeln(A)
```

**المسألة 6:**

**اكتب برنامج يقرأ مبلغ قدره A والفائدة التي يعطيها البنك لهذا المبلغ p والمطلوب طباعة بعد كم سنة يتضاعف المبلغ**

```
Readln(a,p)
Y=0
Value =2*a;
While (a<= value )do
       A= A +A *p;
       Y=Y+1
Writeln(y)
```

**المسألة 7:**

**اكتب برنامج لقراءة عددين a,b وطباعة الأعداد المحصورة بينها تنازلياً**

```
If (a>b)then
    Max =a;  min =b
Else
    max=b;   min =a
first=min ; last =max
while (first <=max ) do
     last = last -1
     fisrt = first +1
     writeln(last)
```

**المسألة 8: اكتب برنامج لقراءة عدد صحيح n وطباعة n! بحيث تعرف**

n!= n*(n-1)*(n-2)*(n-3)*…….1

```
fact: =1;
readln(n)
for i=:1 to n do
    fact=:fact*n;
writeln('The Factor of the number ',n,'is ',fact);
```

**المسألة 9:**

**اكتب برنامج لحساب X مرفوع للقوة n بحيث يكون x,n دخل من المستثمر بطريقة تكرارية**

```
Res =:1;
Readln(x,n)
For i=1 to n do
  Res =Res *X;

Writeln(Res);
```

**المسألة 10:**

**اكتب برنامج لحساب مجموع الأعداد الفردية من 1 إلى 100**

```
Sum=0
For i =1 to 100 do
   Sum = sum +i

Writeln(sum)
```

**المسألة 11:**

**اكتب برنامج لجدول ضرب العدد n بجميع الأعداد من 1 إلى 10**

```
Readln(n)
For i=0 to 10 do
     Prod =n *i
     Writeln( prod)
```

**المسألة 12 :**

**اكتب برنامج يوجد جدول الضرب لجميع الأعداد من 1 إلى 10 وذلك بان نوجد جدول خاص لضرب الأعداد مثلا بين 1 وبين باقي الأعداد وبين 2 وباقي الأعداد وهكذا**

**نحتاج إلى حلقتين كوننا نسير على مجموعتين من الأعداد**

```
For i =1 to 10 do
  For j = 1 to 10 do
     begin
      Prod= i*j
      Writeln(prod)
     end;
```

**المسألة 13:**

**اكتب برنامج يقوم بعملية الضرب باستخدام الجمع**

```
Readln(m,n)
mult=:0;
For i=:1 to n do
      Mult=:Mult+m;
Writeln(mult)
```

**المسألة 14:**

**اكتب برنامج لقراءة وطباعة التاريخ ويقوم بما يلي**

**قراءة متحول يمثل السنة 1989 إلى 2007**

**قراءة متحول يمثل الشهر من 1 إلى 12**

**قراءة متحول يمثل رقم اليوم 1 إلى b**

**بحيث يمكن ان نناقش إذا كانت السنة كبيسة ام لا ونعود b إلى الشهر المناسب فهناك اشهر فيها 31 يوم واخرى 30 و29**

**المطلوب إدخال المستخدم رقم التاريخ وإذا ادخل رقم خاطىء نطلب منه إعادة الإدخال حتى يتم الادخال الصحيح**

**وعند الادخال الصحيح يتم أظهار التاريخ بشكل كامل**

```
Cyear =false    هل هي سنة كبيسة ام لا
Between = false
Read a,b // years
While not (between ) do
    Read (year )
    If (year >=a ) and (year <=b )
        Between =true
        If (year mod 4 =0 )
              Cyear =true  لن ندخل بكثير من التفاصيل سنقبل بالقسمة على 4 فقط ولن نتطرق إلى كافة الشروط
        Between =false
       While not (between )do
               Read(month)
               If (mnth >=1 ) and (month <=12 )
                   Between =true
A=1  , b=31
If (m=2)
         If cyear =true then
               B=29
         Else b=28
Else if (m=4) or (m=6) or (m=9) or (m=11)
             B=30
Between =false
While not (between)do
           Read  (day)
            If day>=1 and day<=b
                    Between =true
Writeln(day,  month, year )
```

**المسألة 15 :**

**اكتب برنامج يقوم بعملية القسمة باستخدام عملية الطرح**

```
Read m,n ;   sub=:1;
         repeat
                   sub  =sub+1;
                   m  =m-n;
         until  (m-n <>0)
  Writeln('The result is ',sub);
```

**المسألة 16:**

**أكتب برنامج يطبع n من الأعداد الموجبة بشرط ألا يطبع الأعداد الأولية:**

**تعريف العدد الأولي : هو العدد الذي يقبل القسمة على نفسه والواحد فقط**

**مثال العدد 7 عدد أولي لانه يقبل القسمة على نفسه وعلى ال1 فقط**

```
Readln(num);      j=0;
While (j <=n )do
     begin
      readln(m);                                          {الأرقام المدخلة من المستخدم}
       isdone_ check=:false;                    i=:2;
        While  (i  <=m-1  )and (not  isdone_ check (do    {true }
          if  i mod m=0 then           { أي انه ليس أولي }
               isdone_ check=:true;
          else
                i=:i+1;
           isdone_ check=:false;
          end; {while  2}
        if  isdone_ check= true then
       write(m);
        j  =j+1;
end;                                      { while 1}
```

**شرح الطريقة**

**مثال لنفرض انه n=3 أي هناك 3 أعداد للإدخال نقرأ العدد m ثم نسند ل i متحول من نمط عدد صحيح العدد 2 لان الواحد يقبل القسمة already وهذه العملية ستبين التالي : نفرض ان تم إدخال 6 ثم 7**

**بالنسبة لل 6 سيدخل i وياخد القيم من 2 –5 وبيشوف إذا في عدد مثل 2 هلق هو يقبل القسمة على 2 إذا هيك بيطلع وما بيطبع العدد بينما إذا فتنا على 7 لح نلاقي انو مافي أعداد من 2-6 بتقبل القسمة على 7 فلا نطبعه**

**المسألة** 17:

**اكتب برنامج يجمع آخر عددين مدخلين**

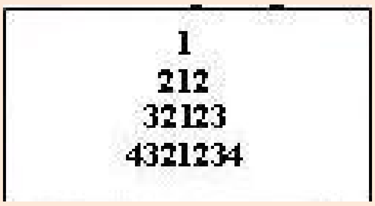
```
Readln  n  ;    عدد الأرقام المدخلة              sum=:0;
for i=:1 to n do
     begin
     Readln m ;
      if i+1=n then
            sum=:sum+m;
      end;
writeln(sum)
```

أو طريقة أخرى نحن لا نعلم عدد الأرقام المدخلة

```
Sum=0,m=0;
C :srting
While ( c <>‘no ,) do
begin
     Writeln(‘Do u want to read a number ‘)
     Readln(c)
     Sum2=m; // The old value of m
     If c=’yes’ then
          Readln(m)
          Sum2=m;
    Else
          Sum =sum1 +sum2;
End;
```

**المسألة** 18 :

**اكتب برنامج يطبع الشكل حيث يدخل المستخدم عدد n ليحدد عدد الأسطر**




```
Readln   n   ;
Writeln(1);
for i=:2 to n do
        for j=:i downto 2 do
                 write(j:4);
        for j=:1 to i do
                 write(j:(4));
```

**المسألة 19:**

**- إكتب برنامج يدخل 10 أعداد تكون محصورة بين 10-100 ويطبع التي لم تتكرر**

**ملاحظة للحل: بما إن لدينا 10 أعداد فنشكل مصفوفة لا ن بعدها ثابت ونضع الأرقام ثم نطبع الأعداد غير المتكررة**

```
var A :array[1..10 ]of integer;
i=:1;
while i<=10 do
begin
read   m
        if   m in 10..100   then    {m >=10   &    m<=100}
                begin
                        A[i]=  m;
                        i=:i+1;
                end;
        else
                writeln(  'Enter a number which have the domain 10..100' )
j=:1;
{طباعة غير المكرر //}
while   j< 10   do
for k=:j+1 to 10 do
begin
        if A[j  ]=  A[k   ]then
        found=:true
        else  found=:false;
end;
if found=false then                        writeln  A[j];
end;
```

**المسألة 20:**

**اكتب برنامج يقوم بتبديل بين أول مرتبة وآخر مرتبة بالعدد**

**مثال : العدد 1217 يصبح 7211 سؤال وظيفة عملي**

```
num,mid=0,n2,digits,last,first,n1=0,res :   integer;
readln num;
first =num mod   10;
n2 =num;            digits =1;
        while( n2 div  10 <> 0 )do
        begin
                Digits = digits +1;
                n2 =n2 div  10;
                last =n2;
        end

n2 =num div 10;
int digit_counter =1;
        for(  i=1 to digits-2  )do
        begin
                mid  =n2 mod  10    *  powf(10,digit_counter);
                n1 =n1+mid;
                n2 =n2 div  10;
                digit_counter=digit_counter+1 ;
        end


for(  m=1 to digits-1  )do
first =first*10;

res =last+first+n1;
Writeln   (res       )
```

**شرح الطريقة** ناخذ الرقم الاول بعملية واحدة هي العدد mod ال10 بينما الرقم الأخير نقوم بحساب عدد خاناته ولنفترض لدينا الرقم 1234 فينتج لدينا 4 خانات فيكون آخر عدد 1234 مقسوم على 1000 فينتج 1 فيكون العدد الناتج هو 4000+ 1 + رقم نشكله وهو يمثل القسم الأوسط من العدد نقوم بعملية بسيطة لنشكل العدد 230 وهذه العملية بعد تعريف digit_counter بالقسم الأوسط من الكود

**المسألة** 21:

**اكتب برنامج يقوم بإدخال عدد من المستثمر ويحسب عدد خاناته**

```
Function Number_digit (n:integer)integer;
Var digit:integer;
Begin
Digit=:0;
While n div 1o <> 0 do
        begin
                digit =:digit+1;
                n=:n div 10
        end;
end;
```

**المسألة** 22:

**اكتب برنامج يقيوم بمعرفة هل الرقم المدخل بدايتو تساوي نهايتو**

```
Readln  n
first=:n mod 10;
while n   <>0 do
begin
        last=:n div 10
        n=:n div 10
end;
if first  =last then                  write  first,last  ;
```

**المسألة** 23:

**المطلوب كتابة برنامج يقوم بجمع الأعداد** m **المتتالية وفق القاعدة التالية مثال:** m=8

8-7+6-5+4-3+2-1=sum

```
Read  m
For k=:1 to n do
    If k mod 2  =0 then
            Sum=:sum-k;
    Else
             sum=:sum+k;
```

**المسألة** 24:

**ندخل عدد الثواني ثم يقوم الحاسب بإظهار عدد الساعات ،الدقائق،الثواني على الشكل التالي** h :m: s

```
program time;
var
hour,min,sec,temp,temp2,day,t:longint;

begin
writeln('enter your time in the form of seconds');
Readln(t);
sec=:t mod 60;
temp=:t div 60;
min=:temp mod 60;
temp2=:temp div 60;
hour=:temp2 mod 24;
day=:temp2 div 24;
writeln('day=',day,' hour=',hour,' min=',min,' sec',sec);
readln;
end .
```

أو يمكن الحل بهذه الطريقة

```
Function getTime(seconds :LongInt :)String;
  Var
    H, M, S :Integer;
  Begin
    H  =:seconds div 3600;
    seconds  =:seconds mod 3600;
    M  =:seconds div 60;
    S  =:seconds mod 60;
    getTime  =:IntToStr(H + )':'  +IntToStr(M + )':'  +IntToStr(S);
  End;
```

**المسألة**25:

**اكتب برنامج ينشى السلسة التالية :**

1-1-2-3-5-8-13-21-34…..

**هذه السلسة هي سلسة فيبوناتشي حيث ينتج كل حد نتيجة جمع حدين سابقين له**

**الحل وفق متحولين وسيطين دون العودية وفق طريقة تكرارية وهي اسرع بكثير من العودية**

```
x=0;
 y=1
for i=2 to n-1 do
  begin
    temp=y
    y=y+x
    x=temp
  end
```

**هذا الحل هو بطريقة تكرارية وسيتم أخذ الطريقة العودية في محاضرة العودية**

**المسألة**26:

**أوجد جذر عدد مدخل موجب**

**الحل يترك كوظيفة للقارىء**

**احدى الطرق تتمثل بان نجمع عدد الأعداد الفردية المكونة للرقم**

**مثال**

**جذر ال**16 **هو** 4 **لان عدد الأعداد الفردية والتي مجموعها يساوي**16 **هي** 4 **أعداد فردية**

7+5+3+1

## بعض الملاحظات الهامة في التعامل مع المحارف في باسكال

**الأعداد :**

**متمم العدد للعدد 9**

```
Ch=:chr(ord(0) + ord(9) - ord(ch));
```

**متمم للعدد 10**

```
Ch=:chr(ord(0) + ord(9) - ord(ch) + 1);
```

**متمم عدد من اجل أي عدد n جميع الحقوق محفوظة لواز**

```
X=:n-9
Ch=:chr(ord(0) +  ord(9) -  ord(ch)  +   x)
```

**لتوضيح المثال نأخد الرقم 0 ونعطي ال ord تبعو قيمة ولتكن 100**

| | |
|---|---|
| 100 | Ord0 |
| ….. | ….. |
| 109 | Ord9 |

**الأحرف**

**التحويل من حرف صغير إلى كبير**

```
a..z
ch=:chr(ord('A')  + ord('a') - ord(ch));
```

**أو يمكن بطريقة أخرى**

```
ch=:chr(ord('Z'  ) - ord('z')  + ord(ch));
```

**التحويل من حرف كبير إلى حرف صغير**

```
A..Z
Ch=chr(-ord('a')  +ord('A') +ord(ch));
```

**المسألة** 1:

**اكتب برنامج لتحويل سلسلة من الكلمات من كبتل إلى اسمول وحساب أحرف العلة**

**تذكرة : أحرف العلة بالإنجليزية تذكرة**a,u,e,o,i

```
S:string
for i=1 to length(s )do
if ord(s[i ]>'A' )    and      ord(s[i]<'Z' )
        s[i ]= chr(ord(c)- ord('A') +ord('a'))

if s[i]= 'a' or s[i] = 'u' or ……. .then
        Vowel=vowel +1
```

**المسالة** 2:

**اكتب برنامج يعكس كل كلمة داخل سلسلة**

```
first,last:integer
s:string[100]
first=1
while i<= length(s )do
if s[i] = '  ' then
    begin
        first=i+1 ; last=i-1
        for j=last downto first do
              write s[i ];
         writeln;
    end;{if}
else
     i=i+1
```

**المسالة 3:**

**اكتب برنامج يأخذ أول حرف من كل كلمة ويطبعه**

```
Write  s[1] : 4
for j=2 to length(s)do
      if s[j] =' ' then
            write(s[i+1] : 4)
```

**المسألة 4:**

**اكتب برنامج يعرف ان هل الكلمة تقرأ من الجانبين ام لا [بكلمة أخرى هل هي متناظرة ام لا بالنسبة للمنتصف**

```
For i =1 to length (s) div2 do
 begin
   If  S[i] = S[length (s) –i+1] then
             Is_semtirc =true;
   Else
           Begin
             Is_semtirc =false ;
             Break;
           End;
  End;
```

**المسألة**5:

**اكتب برنامج تدخل سلسلة من الكلمات ويعكس جميع الكلمات ما عدا اول وآخر كلمة**

إيجاد خانة البداية لثاني كلمة وآخر كلمة

```
For i =1 to  length (s) do
   If S[i] =‘   , then   {space }
          First =i
   If S[length (s) –i+1 ] = ,   , then   {space }
          Last = length (s) –i+1
```

العمل ضمن مجال الكلمة لعكس الأحرف

```
First _temp_word = first;
For i = first to last do
    If S[i] = ‘ ‘ then    {space }
        begin
           Last_temp_word  =    i;
           For j=  First _temp_word to  Last_temp_word div 2  do
                    S[j]= S [Last_temp_word –j +1]
           First _temp_word= i+1;
        End;
```

**المسألة**6:

**اكتب برنامج يطبع أوسط كل حرف من كل كلمة**

```
First=1;
For i =1 to  length (s) do
      If S[i] =‘   , then   {space }
            Begin
                   Last =i;
                   Writeln(S[first+last /2] )
                   First =i+1
              End;
```

**المسألة** 7:

**اكتب برنامج يطبع كلمة كبتل وكلمة اسمو**

```
i=1
for j=1 to length(s)
        if i mod 2=o then  زوجي
              begin
                 if ord(s[i)]>’a’ ) and ord(s[i ]<’z’ )then الحرف صغير
                       write  s[i] : 4
                else
                       write (chr(ord(s[i-)]ord(‘A’+)ord(‘a’ ))
              end {if}
           else فردي
                 if ord(s[i)]>’A’ )and ord(s][i]<’Z’ )then
                           write(s[i:]4);
                   else
                           write (chr(ord(s[i ]+(ord(‘A’ (- ord(‘a’))
```

**المسألة 8:**

**المطلوب إدخال نص [ محرف محرف ] ينتهي بنقطة ومن ثم إظهار النص المدخل و ذلك بعد عكس المحارف**

**مثال: النص المدخل** .dido

**و النص بعد المعالجة** .odid

```
A:array [1..100 ]of char
C:char
Repeat
Read  c
    A[i]   = c
     I=:i+1;
Until c='.'
For j=:I downto 1 do
     Write(A[i:]4)

program paragraph (input,output );
const
  n  =10;
var
  a  :array[1..n ]of char;
   i:integer;
  m:char;
 begin
   read m  ;
   writeln (' Please enter the text in order to reverse it'  );
    i=:1 ;
   while (m<> '  .' )and (i<>n)do
       begin
          a[i =: ]m;
          read  m  ;
          i =:i+1;
       end;
   for i=:n downto 1 do
     begin
       write(a[i )];
     end;
  readln;  readln;
 end.
```

**المسألة** 9:

**اكتب برنامج بلغة باسكال يقوم بقراءة سلسلة من المحارف[string] تنتهي بالمحرف[.] ثم يقوم البرنامج بأيجاد المحرف ذي التواتر الاكبر[المحرف الذي يرد في سلسلة المحارف لاكبر عدد من المرات] وأظهاره.. الحل من قبل ندى الطيان**

```
program test;
type
vect=array['a'..'z']of integer;
var   j:char;     max,i:integer;  s:string;   a:vect;     found:boolean;
```

```
begin
writeln('write the text')
read(s);
i=:1;
j=:'a';
while(s[i]<>'.'do
begin
found=:false;
j=:'a';
      while(j<>'z' ) and(not found)do
       begin
           if(s[i] =j )then
                    begin
                      a[j]= a[j+1]
                       found=:true;
                     end;
             j=:succ(j)
          end;
       i=:i+1;
end;
```

```
(*this section is to count the freq for each letter *)
max=:a['a'];     (*comparing the max *)
for j=:'a' to 'z' do
begin
     if(max<a[j] )  then
       max=:a[j] ;
end;

for j=:'a' to 'z' do  (*showing off that max*)
begin
if(max=a[j] )then
begin
case j of
     'a':writeln('a');
     'b':writeln('b');
     'c':wrietln('c');
.........
     'z':writeln('z');

end;   (*case*)
end;    (*if*)
end;  (*for*)
readln;

end.
```

```
program frexqencychar ;
var   ch:array[1..30]of char ;  i,max,k,n:integer;   maxl,L:char;
begin
i=:1;
read(ch[i]);
max=:0;
while ((ch[1] = '.') do

 {in case of that the first char was '.'}
begin
        writeln('please enter the text again';
         read‘,ch[i]);
end;
 i=:2;
read(ch[i])
while ((ch[i]<> '.' and )i<=30)) do
   begin
        i=:i+1;
        read(ch[i]);
   end;
n=:i;
for L=:'a' to 'z' do
  begin
      k=:0;
      for i=:1 to n do
           if ch[i]=L then
                  k=:k+1;
          if (k>max) then
              begin
                 max=:k;
                maxl=:L;
             end;
     end;

writeln('here is your letter and its frequency :',maxl,' ' ,maxl:4);
readln;
readln;
end.
```

**المسألة** 10:

**أكتب برنامج لفصل سلسلة حرفية الى ثلاث مجاميع واحدة للحروف الكبيرة و الثانية للأرقام و الثالثة للحروف المتبقية من السلسلة الحرفية**

**طبعا باستخدام ال**string

```
Capital :array [‘A’..’Z’ ]of char
Number:array [1..9 ]of integer;
ordinary:array[‘a’..’z’]of char

Writeln(‘Enter the text’);

Read(s);  *(s is verify as string )*;
For i=:1 to length(s )do
Case s[i ]of
‘A’..’Z’    :capital[i] =s[i];
‘1’..’2’     :number [i] =s[i];
‘a’..’z’     :ordinary[i] =s[i];’
```

## الفصل الرابع                                                   العمليات على المصفوفات

**المسألة 1:**

**أكتب برنامج بلغة الباسكال يقرأ مصفوفة أحادية بعدها** n **من النمط** integer **ثم يقوم بإيجاد:**

1-**أكبر عدد في المصفوفة.**

2-**أصغر عدد في المصفوفة.**

3-**المتوسط الحسابي.**

4-**عدد مرات تكرار عدد معين ضمن المصفوفة [يقرأ هذا العدد من قبل المستخدم].**

```
program Vertix ;
var    n,max,min,avg,freq,i :integer;
           A: array[1..50] of integer;
begin

Writeln('Enter the length of the Array ')
Readln(n);
Min=0 ;  max=0;    avg=0; freq=0;
For i =1 to n do
begin
  Read(A[i]);
  If A[i] <min then
     Min =A[i]
  If A[i]>max then
    Max =A[i]
   Avg= Avg+A[i]
End;
Avg= avg / n;
Readln(m)
For i=1 to n do
   If A[i] =m then
       Freq =Freq +1;

Writeln(freq)

end.
```

**المسألة** 2:

**أكتب برنامج بلغة الباسكال يقوم بقراءة مصفوفة أحادية بعدها** n **ثم إيجاد** :

1-**مجموع الأعداد الزوجية في المصفوفة** .

2-**مجموع الأعداد ذات الترتيب الفردي في المصفوفة [ هذا الطلب دون استخدام** mod **]**.

```
program  Vertex ;
var  n,i,sum :integer;
  A: Array [1..100] of integer ;
begin

Readln(n)
Sum=0;
For i =1 to n do
 Readln(A[i]);

For i=1 to n do
  If A[i] mod 2 = 0 then
          Sum = sum+1;
2:
I =1
While i <= n do
     Sum= sum+1
     I=i+2;

end.
```

**المسألة 3:**

**أكتب برنامج بلغة الباسكال يقوم بقراءة مصفوفة** A **و إعادة مصفوفة أخرى** B **بحيث تكون** B **تحتوي عنصر من** A **ثم ضعفه ثم العنصر التالي من** A **ثم ضعفه و هكذا** ....

**مثال :**

A=3,7,2

B=3,6,7,14,2,4

```
program  vertex;
Const max_n=100;

Var  A = Array[ 1.. max_n] of integer;
B =Array [1..2* max_n] of integer;
N:integer;

Begin
     Readln(n);
    For i =1 to n do
       Read(A[i])
I=1;   j=1;
   While i<=n  do
    begin
       B[j]=A[i];
       B[j+1]=A[i]*2;
       J= j+2;
       I =i+1;
     end;

    For i =1 to n do
        write(B[i]),'     ');
end.
```

**المسألة4:**

**أكتب برنامجاً يأخذ مصفوفتين A,B بعد كل منهما n و دمجهما في مصفوفة ثالثة C بطريقتين:**

1- **ملء C بعناصر A كلها ثم عناصر B كلها.**

2- **ملء C بالعناصر بالتناوب واحدة من A و الأخرى من B [و ذلك بطريقتين الأولى باستخدام عدادين و الثانية باستخدام عداد واحد ].**

**مثال:**

A=4,1,9,3

B=2,6,7,5

**في الحالة الأولى:**

C=4,1,9,3,2,6,7,5

**في الحالة الثانية**

C=4,2,1,6,9,7,3,5

**نعتبر ان المستخدم قد قرأ عناصر المصفوفة من قبل المتسخدم لكل من المصفوفتين a,b**

```
For i=1 to n do
  C[i]=A[i];
   C[n+i]=B[i];
1:
i=1
j=1;
while i<= n do
   begin
   C[j]=A[i]
   C[j+1]=B[i];
    i=i+1
    j=j+2
  end;
```

الطريقة الثانية :

```
           k=1 ;
          while (k < =2*n) do
          begin
                    C[ k ]=A[k/2+1];
                    C[k+1]=B[k/2+1];
          k=k+2;
          end
```

**المسألة 5:**

**لدينا مصفوفة مربعة A اكتب برنامجاً لحساب :**

1- **مجموع العناصر الواقعة فوق القطر الرئيسي [ مع عناصر القطر الرئيسي ] .**

2- **مجموع العناصر الواقعة تحت القطر الرئيسي [ مع عناصر القطر الرئيسي ].**

**ملاحظة ..**

**لحل المسألة يستحسن رسم مصفوفة و كتابة أرقام الخانات التي يجب جمعها على المسودة ثم مقارنتها للوصول إلى الحل المناسب .**

**فمثلاً عندما أريد حل الطلب الأول**

**أكتب**

□ **عندما** n =5

i=1-->j=1,2,3,4,5

i=2-->j=2,3,4,5

i=3-->j=3,4,5

i=4-->j=4,5

i=5-->j=5

□

□ **عندما** n=4

i=1-->j=1,2,3,4

i=2-->j=2,3,4

i=3-->j=3,4

i=4-->j=4

□

**لاحظ المجال الذي تأخذه ال** i ....

**ثم لاحظ مالذي يربط بين كل قيمة ل** i **و المجال الذي تتراوح بينه ال** j **من أجل هذه القيمة ؟؟؟؟**

```
     A[10,10]  :Matrix
      n   :integer;
     writeln('Input The Length of A');
     readln( n) ;
     writeln( 'Input The element of A ...';
     for i= 1 to n do
     begin
         for j= 1 to n do
         begin
             readln A[i,j];
         end;
     end;
     sum =0;
{calculation The Sum of the elements of matrix That Upper Main diameter}

writeln('calculation The Sum of the elements of matrix That
Upper Main diameter  ..') ;
     writeln('The Index of The element ..') ;
     for  i=1 to n do
     begin
         for j=i to n do
         begin
             sum =sum  +A[i,j];
             writeln('i =', i,'j"= ', j );
         end;
         writeln;
     end;
     writeln('The Sum " : ', sum );

{calculation The Sum of the elements of matrix That under Main diameter }
writeln('calculation The Sum of the elements of matrix That
under Main diameter ');
     sum =0;
     writeln('The Index of The element...') ;
     for i=1 to n do
     begin
         for j= 1 to i do
         begin
             sum =sum + A[i,j];
             writeln('i" =', i,'  j"= ', j );
         end;
         writeln;
     end;

     writeln('The Sum " : ', sum );
```

**المسألة6:**

□**اكتب برنامج يعكس عناصر القطر الرئيسي أو الثانوي بمصفوفة ثنائية**

| | | | |
|---|---|---|---|
| **1** | 2 | 223 | 22 |
| 35 | **2** | 26 | 24 |
| 45 | 5 | **3** | 27 |
| 54 | 54 | 54 | **4** |

□**تصبح على الشكل**

| | | | |
|---|---|---|---|
| **4** | 2 | 223 | 22 |
| 35 | **3** | 26 | 24 |
| 45 | 5 | **2** | 27 |
| 54 | 54 | 54 | **1** |

**ملاحظة قبل الحل**

**المصفوفات المربعة هي فقط التي يوجد لها قطر رئيسي او ثانوي حيث المصفوفة من البعد** n

**شرط القطر الرئيسي هو** A[i]=A[j] أو A[i,i] و **الثانوية** i+j=n

```
For i =1 to n do
      temp= A[i,i]
      A[i,i]= A  [n-i+1 ,n-i+1 ]
      A  [n-i+1 ,n-i+1 ] =temp;

       Swap(A [i,i], A  [n-i+1 ,n-i+1 )]
```

**المسألة 7 :**

**اكتب برنامج يبدل بين أقطار مصفوفة أي يجعل القطر الرئيسي ثانوي والثانوي رئيسي**

```
for i=:1 to n do
    temp= A[i,i]
    A[i,i]= A   [i , n+1-i]
    A   [i , n+1-i]     =temp

       swap A[i  , i ]  ,    A   [i , n+1-i]
```

**المسألة 8:**

**لدينا مصفوفة ثنائية البعد تدخل الرقم المراد البحث عنه num وكم مرة إذا تكرر ويطبعه إذا تحققت الشروط إن كان عدد التكرار freq أكبر من قيمة max يدخلها المستخدم**

```
Read(num,max);
freq=:0;
for i=:1 to n do
for j=:1 to n do
       begin
       if A  [ i  ,  j  ]  =num   then
       freq=:freq+1;
       end;
if  (freq>max) then
 write(freq)
```

**المسألة 9: إكتب برنامج يقوم بتبديل أصغر عنصر في مصفوفة بأكبر واحد وإعطاء إحداثيات العنصر الأكبر والأصغر قبل النقل**

```
Max=[ a , b];   min=[c, d]
max=:A[1,1];     min=:A[1,1];
for i=:1 to n do
for j=:1 to n do
begin
       if A[i,j]>  max then
              begin
                     max=:A[i,j];    a=:i;  b=:j;
               end;
       if A[i,j  ]<   min then
              begin
                     min=:A[i,j];    c=:i;   d=:j;
              end;
end;
swap  (A[a,b ]    ,  A[c,d)  ]
Writeln 'Max is',max'min is',min
Writeln( 'The position for max is',a,b,'And for the min ',c,d)
```

**المسألة 10:**

**اكتب برنامج يزيح مصفوفة ثائية إلى اليمين بمقدار** k

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

k=1 **تصبح وعندما تساوي 3 نرجع للمصفوفة الاولى**

| | | |
|---|---|---|
| 3 | 1 | 2 |
| 6 | 4 | 5 |
| 9 | 7 | 8 |

**نقوم بنقل الأرقام من كل عمود ومن ثم نصفر آخر الأمر**

```
if    k>n then        writeln   'Error'   ;
shift_counter =0;
else
         begin
                for  raw =1 to n do
                       B [raw]  = A   [raw ,   n  ]
               while (  shift_counter < k   )do
                        for j =:n downto 2 do
                        for i=:1 to n do
                               A[i,j]   = A[i,j-1];
               shift_counter  =shift_counter +1
               for i=:1 to n do
                      A   [i  ,  1 ]=  B   [i  ]
        end;
```

**مفتاح الحل:**

**نقوم بعدد** K **من العمليات بحيث تتضمن كل عملية نقل عناصر المصفوفة إلى العمود الذي يسبقها بحيث ننقل في آخر الأمر عناصر العمود الأخير إلى العمود الأول**
**وبتكرار العملية عدد** k **من المرات فينتج لدينا المصفوفة المزاحة**

**المسألة 11:**

**- إكتب برنامج يدخل مصفوفة أعداد أحادية شرط ألا تتكرر القيم (سؤال دورة)**

**الحل :**

```
i=0
while  i<=n
begin
      Read  A[i]
      for j=1 to i-1 do
         if A[i ]= A[j]then
            i=i
         else
            i=:i+1
end;
```

**المسألة 12:**

**إكتب برنامج يبدل عناصر المصفوفة الأحادية المواقع الزوجية مع المواقع الزوجية**

**مثال**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 8 | 33 | 57 | 86 | 632 | 1 | 19 |

تصبح

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 8 | 2 | 57 | 33 | 632 | 86 | 19 | 1 |

```
i=1
while  i<=n
  if A[i]mod 2=0   then
      swap(A[i ], A[i+1 ])
```

مع مناقشة حالة إذا كانت آخر خانة حيث نطرح 1 بدل من الجمع

```
  i=:i+2
```

**المسألة 13:**

**مصفوفة ذات بعدين إطبع الأرقام وكم مرة تكررت بدون تكرار الأرقام**

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 5 | 6 | 8 | 910 |
| 23 | 82 | 9 | 787 | 8 | 54 | 2 |
| 0 | 910 | 8 | 6 | 5 | 1 | 1 |
| 82 | 9 | 8 | 2 | 2 | 5 | 1 |

**الحل : نعتبر هناك تابع مساعد يدعى freq ندخل لها عدد ومصفوفة وترد لنا عدد تكرار هذا العدد ضمن المصفوفة**

```
function freq (A:Array; num :integer)integer
          frq=:0;
          for i=:1 to n do
          for j=:1 to n do
                    if A[i,j] n=n num
                              frq=:frq+1
          freq  =frq
end;    { func}

for i=1 to n do
   for j=1 to n do
         writeln(    A[i,j]:4,'          '  ,freq(A[i,j]))
```

**سيحدث تكرار لبعض القيم لمنع التكرار نضع مصفوفة إحادية مساعدة Aux لمعرفة ان تكرر العنصر ام لا ان تكرر لا داعي لحسابه مرة أخرى**

**بحيث نضع كل رقم ضمن المصفوفة الاحادية بحيث يكون بعدها بأسوأ الاحوال n*n وذلك في حالة كانت تحوي ارقام لا تملك تكرارات وقبل ان اطبع تكرار العدد انظر بالمصفوفة المساعدة هل الرقم الذي امر عليه موجود ضمنه ام لا إن كان موجود فلاداعي لكتابته**

```
for i=1 to n do
   for j=1 to n do
       begin
             for k=1 to n*n
                if Aux[k]= A[i,j] then
                 begin
                 found =true
                  break
                 else {do nothing}
               if not found then
                        writeln(    A[i,j]:4,'          '  ,freq(A[i,j]))
       end;
```

**المسألة 14:**

**اكتب برنامج يقوم بترتيب عناصر أعمدة المصفوفة الثنائية تصاعدي او تنازلي المصفوفة مربعة**

**ملاحظة:سنكتب طريقة تعئبة المصفوفة لكي يظهر للقارىء طريقة الترتيب**

```
{i lines und j columns}
for i=:1 to n do
  for j=:1  to n do
      read(A[i,j];)

for j=:1 to n do
   for i=:1 to n-1 do
        if A[i+1 ]> A[i ]then
           begin
             temp=:A[i+1];
             A[i+1]=:A[i];
             A[i]=:temp;
           end
```

**المسألة** 15:

**مصفوفة أحادية حولها إلى مصفوفة ثنائية البعد وفق الشكل التالي**

```
1234567
1230567
1200067
1000007
```

```
Print  A
        if n فردي
mid1=mid2 =(  n div2) + 1
        else
        mid1=n div 2  ;
         mid2=(n div2)+1
writeln;
   A[mid1]=  0;
 A[mid2]=:0;
print(A) {اطبع المصفوفة مرة أخرى}
for j=3 to n
    A[mid1+1]= 0;
    A[mid2-1]= 0;
Print   (A)
```

التابع print يقوم بطباعة المصفوفة

**المسألة 16:**

**- إكتب برنامج بلغة باسكال يقوم بقراءة n عدد وتخزينها ضمن مصفوفة[ a ] , والمطلوب:**

**1-كتابة برنامج يقوم بالتالي:في حال كان عدد الرقام المتتالية والمتماثلة[ في المصفوفة a ] اكبر او تساوي[ 2 ] يتم نقل هذه العناصر من المصفوفة[ a ] الى المصفوفة[ b ],**

a:122557776955332

b:25753

**2-كتابة برنامج يقوم بالتالي:نقل العناصر المكررة لاكتر من مرة والغير مكررة في المصفوفة a الى المصفوفة, c لمرة واحدة فقط،اي كالتالي:**

a:1225577776955332

c:125769532

```
readln(n)
for i =1 to digit(n) do   {digit is a fun calculate the num of n digit}
  begin      num=n mod 10 ;   A[i]=num;     n=n div 10;
   end;
j=1;  i=1;  index =1;
1:While i<n do
        If a[i]=a[i+1]then
             begin
                   for k= i+2 to n do
                         if A[k] =A[i] then
                              Index =k;
                         else break;
                  if index >i+1 then begin
                          B[j] =A[i]
                            j=:j+1;
                           i=index+1;   end;
                   else    begin
                          B[j] =A[i]
                            j=:j+1;
                            i= i+2;     end;
              End;
         else                 i=:i+1;
2:
```

نفس الحل السابق ولكن بعد else i=i+1 نضع العنصر A[i] في العنصر B[j] ونزيد عداد j بمقدار واحد

## الفصل5 مسائل عودية

**المسألة 1 : مسألة فيبوناتشي تنص هذه المسألة على التالي :**

f(n) = f(n-1) + f(n-2)
f(0) = 0
f(1) = 1

حل المسألة :

**نلاحظ هنا أنه بإدخال قيمة n صحيحة موجبة يتم حل مسألة فيبوناتشي حسب القيم السابقة لها، فمن أجل n=3 يكون لدينا:**

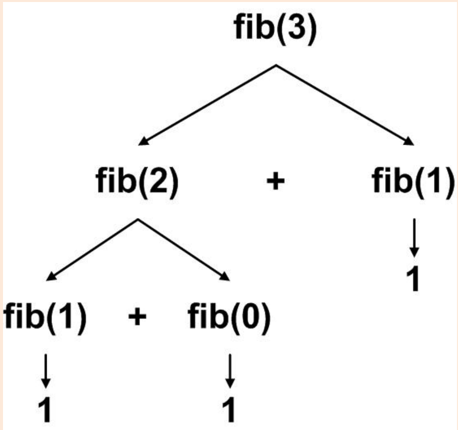
fib(3) = fib(2) + fib(1)

fib(2) = fib(1) + fib(0)

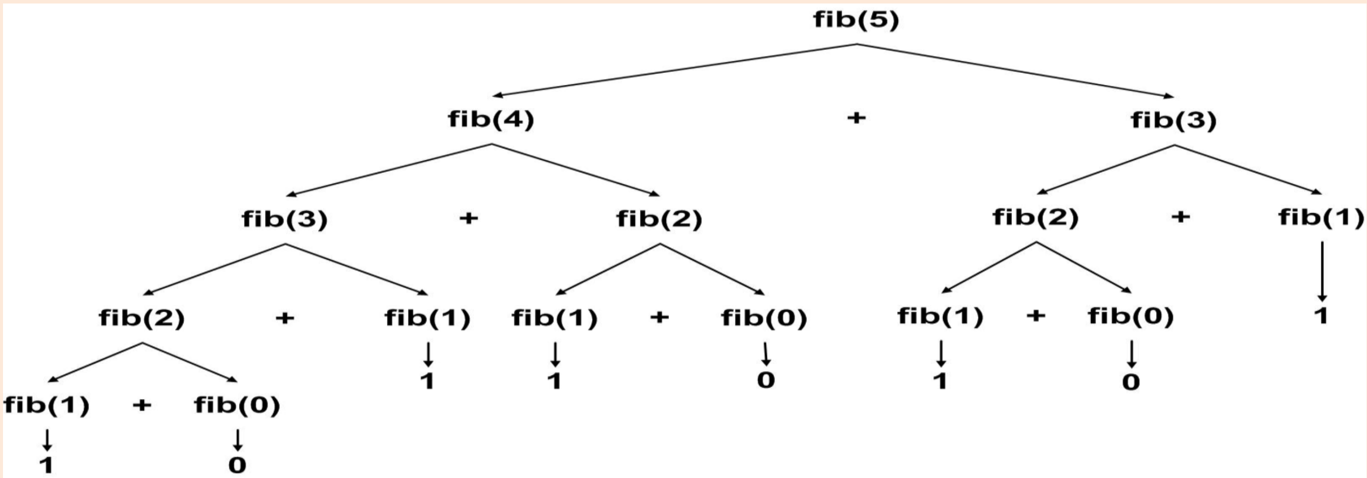
==> fib(3) = fib(1) + fib(1) + fib(0) = 1 + 1 + 0 = 2

==> fib(3) = 2

**يمكن تمثيل شجرة فيبوناتشي بالشكل:**




**ومن أجل N=5 يكون**



```
function Fib(n : integer) : integer;
begin
  if n = 0 then
    Fib := 0   { return the value Fib(0) = 0 }
  else
    if n = 1 then
      Fib := 1   { return the value Fib(0) = 0 }
  else
    Fib := Fib(n-1) + Fib(n-2);
end;   { end of function }
```

**<u>المسألة 2</u> : <u>مسألة فيبوناتشي باستخدام الشعاع</u> :**

**هنا سنقوم بالحصول على قيمة فيبوناتشي .. لكن عن طريق شعاع يحوي قيم فيبوناتشي من أجل القيم من [ 1 ] إلى [ 10 ].**

**الغرض من الشعاع هو تسجيل قيمة فيبوناتشي من أجل كل N .. بحيث لا نقوم بكل مرة بحسابها عودياً.**

**تعريف الشعاع :**

```
const
  Max := 10;   { defining the max size of Vector }
type
  Vector = array [ 2 .. Max ] of integer;   { defining Vector }
var
  a : Vector;   { defining an array variable from Vector }
```

**لم نأخذ هنا المجال من الواحد لأن قيمة N=1 معروفة لنا وهي مساوية للواحد حيث**

fib(1) = 1

**نقوم هنا بالبحث عن قيمة N والتي تعبر عن مفتاح النسق Vector لأننا سنقوم هنا بتخزين القيم كما يلي:**

a[2] = fib(2)

a[3] = fib(3)

a[4] = fib(4)

…

a[10] = fib(10)

**خوارزمية الحل:**

**يتم البحث عن قيمة N ضمن الشعاع a فإن وجد نقوم بإرجاعه للمستدعي، وإن لم يوجد نقوم بحساب فيبوناتشي N كما في الطريقة السابقة، ويتم في كل استدعاء عودي تنفيذ نفس الأسلوب .. التقصي في الجدول ثم الاستدعاء والحساب إن لم يوجد**

**لمعرفة إن كانت خانة القيمة فارغة أم لا، نقوم بوضع قيمة إبتدائية [ الصفر ] في الخانة دلالة على أنه لم يتم حساب القيمة بعد، لذلك نقوم بداية بتصفير الشعاع:**

```
for i := 1 to Max do
  a[i] := 0;   { Zeroing the slots of array }
```

```
function VecFib(a : Vector; n : integer) : integer;
begin
  if n = 0 then
      VecFib := 0
  else
    if n = 1 then
        VecFib := 1
  else
   begin
      if a[n] = 0 then
         a[n] := VecFib(n-1) + VecFib(n-2);
      VecFib := a[n];
   end
end;   { end of function }
```

**المسألة 3 : حساب قوة عدد**

**نريد حساب قوة ( N ) لعدد ( X ) وفقاً للقانون التالي:**

$$x^n = x^{(n\,mod\,2)} * x^{(n\,div\,2)} * x^{(n\,div\,2)}$$

$$x^0 = 1$$

$$x^1 = x$$

**وذلك من أجل** x,n>0

```
function Power( x , n : integer) : integer;
begin
  if n = 0 then
    Power := 1
  else
    if n = 1 then
      Power := x
  else
    Power := Power(x,n mod 2) * Power(x,n div 2) * Power(x,n div 2);
end;   { end of function }
```

الطريقة 2 طريقة عودية أكثر تبسيطا

$$x^n = x^{n-1} * x^{n-2} * .........x^0$$

```
Function Power (x,n:integer):integer;
        if n =0 then
                Power=:1;
        else
                if n=1 then
                        Power=:x
                else
                        Power=:x*power(x,n-1);
```

**المسألة 4 : جمع الأعداد بالتناقص مقدار واحد :**

AddByOne (n) = n + (n-1) + (n-2) + ... + 2 + 1

**خوارزمية الحل :**

**يتم فحص قيمة n إن كانت مساوية للواحد نقوم بإرجاعه واحداً للمستدعي، وإلا فإننا نقوم بجمع n مع استدعاء عودي جديد من أجل n-1**

```
function AddByOne( n : integer) : integer;
begin
  if n = 1 then
    AddByOne := 1
  else
    AddByOne := n + AddByOne(n-1);
end;   { end of function }
```

**مسألة 5 : جمع الأعداد بإنقاصها إثنين :**

AddByTwo (n) = n + (n-2) + (n-4) + .... + ( one or zero )

**يتم هنا جمع الأعداد بدء من n ويتم في كل مرة إنقاصه مقدار [ 2 ]، هنا علينا الإنتباه بأن القيمة الأخيرة في عملية الجمع قد تكون [1 ) أو [0 ] وهذه بعض أمثلة:**

AddByTwo (5) = 5 + 3 + 1 = 9

AddBy (8) = 8 + 6 + 4 + 2 + 0 = 20

**خوارزمية الحل :**

**مشابه للمسألة السابقة .. لكن الفرق يكمن في أنه بدل أن يكون شرط التوقف ( n = 1 ) يكون الشرط هو ( n < 1 ) ويتم هنا إرجاع قيمة [ 0 ] وليس واحد.**

**والاختلاف أيضاً في الاسترجاع العودي حيث نقوم باستدعاء التابع عودياً من أجل ( n-2 )**

```
 function AddByTwo( n : integer) : integer;
begin
  if n < 1 then
    AddByTwo := 0
  else
```

```
    AddByTwo := n + AddByTwo(n-2);
end;  { end of function }
```

**مسألة 6 : جمع عددين عودياً**

**في مسألتنا هذه نقوم بالجمع العادي .. لكن بشكل عودي حيث نقوم بكل استدعاء بالإضافة واحداً ( +1 ) إلى أحد المعاملين وإنقاص الآخر ( -1 ) ويتم نفس التكرار حتى يصبح المعامل التي يتم الإنقاص منه مساوياً الصفر، حينها يكون المعامل الأول هو ناتج الجمع.**

**مثال :**

5 + 3 ==> x = 5 , y = 3

1st recursive call ==> x = 5 + 1 , y = 3 - 1 ==> x = 6 , y = 2

2nd recursive call ==> x = 6 + 1 , y = 2 - 1 ==> x = 7 , y = 1

3rd recursive call ==> x = 7 + 1 , y = 1 - 1 ==> x = 8 , y = 0

x = 5 + 3 = 8

```
function RecAdd( x , y : integer) : integer;
begin
  if y = 0 then
    RecAdd := x
  else
    RecAdd := RecAdd(x+1,y-1);
end;  { end of function }
```

**المسألة 7 : طرح عددين عودياً**

**يتم هنا إنقاص كل من x و y بشكل عودي بمقدار واحد ( -1 ) ويستمر الطرح حتى يصل المعامل ( y وحصراً y ) إلى الصفر، حينها يكون ناتج الطرح هو المعامل x.**

**مثال [ 1 ]:**

5 - 2 ==> x = 5 , y = 2

1st recursive call ==> x = 5 - 1 , y = 2 - 1 ==> x = 4 , y = 1

2nd recursive call ==> x = 4 - 1 , y = 1 - 1 ==> x = 3 , y = 0

x = 3 = 5 – 2

**مثال** [ 2 ] : 3 - 4 ==> x = 3 , y = 4

1st recursive call ==> x = 3 - 1 , y = 4 - 1 ==> x = 2 , y = 3

2nd recursive call ==> x = 2 - 1 , y = 3 - 1 ==> x = 1 , y = 2

3rd recursive call ==> x = 1 - 1 , y = 2 - 1 ==> x = 0 , y = 1

4th recursive call ==> x = 0 - 1 , y = 1 - 1 ==> x = -1 , y = 0

x = -1 = 3 - 4

```
function RecSub( x , y : integer) : integer;
begin
  if y = 0 then
    RecSub := x
  else
    RecSub := RecSub(x-1,y-1);
end;   { end of function }
```

**<u>مسألة 8 : ضرب عددين عودياً</u>**

**وذلك من أجل x,y أعداد صحيحة موجبة**

**يتم هنا الجمع بشكل متكرر .. ويكون الجمع بحيث نجمع قيمة x لنفسها في كل مرة وإنقاص y بمقدار واحد، ويستمر الاستدعاء العودي لحين يصبح y = 1، في حال تم إدخال أحد القيمتين مساوية الصفر، يتم إرجاع قيمة صفر وهي ناتج ضرب أي عدد بعنصر الصفر الماص.**

```
 function RecMult( x , y : integer) : integer;
begin
  if ( x = 0 ) or ( y = 0 ) then
    RecMult := 0
  else
    if y = 1 then
      RecMult := x
  else
    RecMult := x + RecMult(x,y-1);
end;   { end of function }
```

**مسألة9: قسمة عددين عودياً [ كما في تعليمة div ]**

من أجل x,y أعداد صحيحة موجبة، y<x

يتم هنا إنقاص قيمة y كل مرة من x ويتم عدد مرات الإنقاص في الاستدعاء حتى يصبح x<=0 فيتم إرجاع القيمة صفر.

في حال كانت القيمة y مساوية الصفر يكون ناتج القسمة مرفوض، حيث أنه عبارة تقسيم على صفر.

```
function RecDiv( x , y : integer) : integer;
begin
   if x <= 0 then
      RecDiv := 0
   else
      if y = 0 then
         writeln('INVALID DIVISION : Division by Zero')
   else
      RecDiv := 1 + RecDiv(x-y,y);
end;   { end of function }
```

**المسألة10:**

اوجد تابع المضاعف المشترك الأكبر GCD من Ray

الحل موجود بالكتاب وهذه بطريقة عودية أخرى

```
Function GCD(A,B  :integer  ): integer
   • If (A  =B ) then
         GCD  =A
   • Else
          If  (A > B  )then
               GCD  =GCD  (A-B , B )
         Else
               GCD  =GCD  (A , B-A )
```

**المسألة 11:**

**-الفرز بالعوم عودياً**

```
procedure sort(k,n:integer;var a:vector)
var
t:integer;
begin
   if   k=n    or    n=0    then
   begin
        if    n<>0         then
          sort(n-n+1,n-1,a)
        else
          t=:1;
   end
   else if a[k  ]>a[k+1   ]then
        begin
           t=:a[k];
           a[k  ]=  a[k+1]
           a[k+1  ]=   t
           sort(k+1,n,a)
        end
        else
           sort(k+1,n,a)
end;
```

**المسألة 12:**

1: **حساب عاملي عدد** n!=n*(n-1)*(n-2)*……….1

```
Function Factor (n:integer:)integer;
If n<=1 then
           Factor=:1;
Else
          factor=:n*factor(n-1);
```

**المسألة** 13:

حقق تابع إكرمان

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1,1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m,n-1)) & \text{if } m > 0 \text{ and } n > 0. \end{cases}$$

```
Function A (m,n:integer:)integer;
Begin
If m=0 and n>0 then
        A(m,n ) = n+1
Else if m>0 and n=0 then
         A(m-1,1);
If m>0 and n>0 then
         A(m-1,A(m,(n-1)))
```

$$\begin{aligned} A(1,2) &= A(0, A(1,1)) \\ &= A(0, A(0, A(1,0))) \\ &= A(0, A(0, A(0,1))) \\ &= A(0, A(0,2)) \\ &= A(0,3) \\ &= 4 \end{aligned}$$

$$\begin{aligned} A(4,3) &= A(3, A(4,2)) \\ &= A(3, A(3, A(4,1))) \\ &= A(3, A(3, A(3, A(4,0)))) \\ &= A(3, A(3, A(3, A(3,1)))) \\ &= A(3, A(3, A(3, A(2, A(3,0))))) \\ &= A(3, A(3, A(3, A(2, A(2,1))))) \\ &= A(3, A(3, A(3, A(2, A(1, A(2,0)))))) \\ &= A(3, A(3, A(3, A(2, A(1, A(1,1)))))) \\ &= A(3, A(3, A(3, A(2, A(1, A(0, A(1,0))))))) \\ &= A(3, A(3, A(3, A(2, A(1, A(0, A(0,1))))))) \\ &= A(3, A(3, A(3, A(2, A(1, A(0,2)))))) \\ &= A(3, A(3, A(3, A(2, A(1,3))))) \\ &= A(3, A(3, A(3, A(2, A(0, A(1,2)))))) \\ &= A(3, A(3, A(3, A(2, A(0, A(0, A(1,1))))))) \\ &= A(3, A(3, A(3, A(2, A(0, A(0, A(0, A(1,0)))))))) \\ &= A(3, A(3, A(3, A(2, A(0, A(0, A(0, A(0,1)))))))) \\ &= A(3, A(3, A(3, A(2, A(0, A(0, A(0,2))))))) \\ &= A(3, A(3, A(3, A(2, A(0, A(0,3)))))) \\ &= A(3, A(3, A(3, A(2, A(0,4))))) \\ &= A(3, A(3, A(3, A(2,5)))) \\ &= \ldots \\ &= A(3, A(3, A(3,13))) \\ &= \ldots \\ &= A(3, A(3,65533)) \\ &= \end{aligned}$$

**المسألة** 14:

**-حساب اصغر [ اكبر ]قيمة لمصفوفة بطريقة عودية**

```
{out side the procedure }
Read(n);
For j=:1 to n do
Read(A[j]);
Min=:a[n]
Smallest(n,A[n],min);

6:Function smallest (n:integer; A:array of
nteger,min:integer):integer;
If n= 1 then
 begin
       If A[1]<min then
              A[1]=min
       else
           Smallest=:min;
  end;
Else
         If A[n] <min then
              min=:A[n]
smallest (n-1,A[n],min);

smallest=:min;
```

**المسألة 15 :**

**البحث عن قيمة ضمن مصفوفة إحادية**

```
Function linersearch (A  :matrix ; N  :integer ; X  :integer ):Boolean
Begin
  If ( A[N] = X ) then
       B  =true
  else
      If ( N  =0 )and ( A[N] <> X )    then
        B  =false
      else    linersearch ( A , N-1 , X )
  Linersearch  =B
end;
```

**المسألة 16 : البحث الثنائي :**

**يعتمد البحث الثنائي ان هناك مصفوفة إحادية مرتبة بشكل تنازلي أو تصاعدي شرط الترتيب مهم جداً بحيث يمكن ان إختصر عمليات المقارنة الكثيرة وذلك بالبحث بالمنتصف إذا كان الرقم الذي نبحث عنه اكبر من القيمة الحالية ستكون القيمة موجودة بالقسم الإيمن من المصفوفة والعكس بالعكس**

```
Function linersearch (A  :matrix ; first,last  :integer ; X  :integer ):Boolean
begin
mid = first +last div 2;
       if  A[mid] =x then
            linersearch = true
        else if A[mid] > x then
             linersearch(A,mid+1,last,x)
       else
              linersearch(A,first  ,mid,x)
end;
```

**المسألة 17: حساب متوسط عناصر مصفوفة عودياً**

```
Function Average  (A  :matrix ; N:integer :)real ;
Begin
     If  (N  =1   )then
          Sum  =A[N]
     Else
         Sum  =Sum ( A , N – 1 ) + A[N]
          L  =L  +1
     Sum  =Sum /L
end
```

**المسألة 18: حساب مجموع عنصار نسق**

```
Function Sum ( A  :matrix ; N  :integer ):integer
Begin
   If  (N  =0  )then
      Sum  =0
    Else
        Sum  =Sum  (A , N-1 ) + A[N]
```

## الفصل6 مسائل الأعداد متقدم Advanced Numeric

### المسألة1:

اكتب برنامج تدخل له عدد ويحسب عدد تكرارات كل رقم منه مثال العدد1122123 يعطي 1 مكرر 3 مرات و 2 مكرر 3 مرات و 3 مكرر مرة واحدة فقط

الحل يوجد لدينا تابع بداية اسمه pow[x,y] مهمته إيجاد $x^y$ ويمكن تحقيقه بسهولة

```
n  ,    m=0:                                :long ;
num,   temp_n                               :integer;
is_no_frq                                   :boolean;
i,    digit_counter   ,  x                  :integer;

readln  n ;
is_no_frq=true;
digit_counter =0;
m=0;
while n div  10 <> 0  do
    begin
        i=0;
        num =n mod  10;
        temp_n =n;
            while  temp_n  <> 0  do
            begin
                if  num   = temp_n mod 10  then
                     i  =i+1
                else
                begin
                    x  =temp_n mod 10 ;
                    m  =m +x *pow)10.0,digit_counter(;
                    digit_counter =digit_counter +1 ;
                    is_no_frq =true;
                end
                    temp_n  =temp_n div  10;
            End {while 2}
            if  temp_n =0  then
   writeln ('The number of Frequency for ',num,'is  ',i );
            if not  is_no_frq  then
                break تعليمة تقوم بالخروج من الحلقة //
            else
            begin
                is_no_frq=false;
                digit_counter =0;
                n =m;
                m=0;
            end {else }
    End {while }
```

**شرح البرنامج :**

**الرقم n هو العدد المدخل الطريقة تعتمد ان ناخذ اول رقم من العدد n ونبحث عن تكرار هذا الرقم لهذا إحتجنا إلى رقم آخر m كي نضع فيه كل الأرقام غير المكررة كما في مثالنا في نص السؤال بعد اول خروج من حلقة ال while الثانية يكون m=112212 وفي المرة الثانية يصبح m يساوي 111 وبالتالي إستطعنا معرفة عدد تكرار الأرقام ضمن العدد الواحد .**

**المتحول : is_no_frq كي تعبر عن إذا كنا قد إنتهينا ولم نجد عناصر أخرى كي لا تصبح الحلقة لانهائية . يمكن بان نقول العدد بداية إصبح 111 كما في مثالنا السابق وبعد ذلك فهو 0 وبالتالي عندما يصبح 0 فنحن قد إنتيهنا وهذا المتحول يصف متى نصل إلى هذه الحالة**

**المتحول : digit_counter رقم مساعد من أجل تكوين العدد m .**

**المتحول: temp_n كي لا تضيع قيمة n ونضعه بحلقة ال while الثانية**

**التابع :pow هو تابع يحسب الرقم الأول مرفوع بالقوة الرقم الثاني ويمكن تحقيق هذا التابع بسهولة**

```
Function pow (x :integer ; y :integer )
Var res :integer ;
Res =1;
                For i =1 to y do
   Res   =Res   *X;
Pow =res ;
end;
```

**المسألة 2:**

**اكتب برنامج لطباعة مضاريب عدد مدخل مثال عند ادخال العدد126 يظهر لنا2*3*3*7 سؤال دورة 8 علامات**

| | |
|---|---|
| 126 | 2 |
| 63 | 3 |
| 21 | 3 |
| 7 | 7 |

```
function primeFactors (n:integer  )    :   integer
begin
        i=:2;
        while i<=n do
                while n mod i= 0 do
                                writeln(i‘*’);
                                n =n div i;
                i =i+1;
End;
```

**المسألة 3:**

**اكتب برنامج يرتب العدد 98041 ليصبح 98410**

**و 201056 تصبح 215600**

**توضيح للبرنامج يطلب إزاحة الخانات عند وجود صفر فمثلا 8888 يصبح 8888 و 5501 يصبح 5510**

**وإن كان هناك أكثر من صفر فإننا نقرب للصفر الأول مثال 98052013 يصبح 98052130**

```
num ,z,res=0,digit-=1,n2,how_many_zeroz=0 :  long;
     readln (num  );
     n2  =num;
     while ( num <>  0  )do
     begin
          z  =num mod  10 ;

          if(  z  =0 )  then
          begin
               how_many_zeroz  = how_many_zeroz+ 1 ;
               num  =num div  10;
               continue ;
          end;
          else
          begin
               Digit =digit +1;
               res =res  +z  *powf(10,digit);

          end;
          num  =num div  10;

     end;

     if( how_many_zeroz <>  0 )  then
          for i =0 to how_many_zeroz  do
               res  =res  *10;

writeln   (res );
```

**شرح الطريقة :**

12301 **ناخذ العدد خانة خانة ونزيد عداد للخانات وفي حال وجدنا صفر لا يزداد هذا العداد بحيث ينضرب** 1 **ثم** 3 **ثم** 2 **ثم** 1 **بالاعداد** 1 **و** 10 **و** 100 **و** 1000 **على الترتيب في نهاية الأمر يتشكل لدينا العدد** 1231 **ثم نضرب العدد ب** 10 **قوة عدد الأصفار**

## المسألة4:

## اكتب برنامج عملية طرح تقليدية

```
3333
-
213
----
3120
```

**بفرض ان num1 هو العدد الاكبر**

```
num1,num2,n,fdigit1,fdigit2,res=0 :   integer
n --=1;
readln (num1,num2);
     while(num1 <>  0 )do
     begin
               fdigit1  =num1 mod 10;
               fdigit2  =num2 mod 10;
          if( num2 <>  0)then
               begin
               n = n+1;
               if(  fdigit1 > =fdigit2)  then
                    res  = res (+fdigit1-fdigit2 *)powf(10,n );
               else      {  مجاورة خانة من استعارة}
                         begin
                              fdigit1  =+10;
                         res((=fdigit1-fdigit2 *)powf(10,n +(res);
                              num1 =num1 -powf(10,1 );
                         end
                                                  } يـتم تـنفيـذ الـتالي بـالحالـتين{
                    num1=num1 div  10;
                    num2 =num2 div  10;
               end
          else
               begin
                    n =n+1;
                    res =+fdigit1 *powf(10,n );
                    break;
               end
     end
     writeln ('The result is ',res )
```

**المسألة 5:**

**نفس البرنامج 21 لعملية الضرب والجمع**

**الجمع**

```
num1,num2,n,fdigit1,fdigit2,res=0,rem1=0,rem2=0 : integer;
n-=1;
readln(num1,num2);
     while  (num1 <>  0) do
     begin
               fdigit1  =num1  mod 10;
               fdigit2  =num2 mod  10;

          if ( num2 <>  0) then
               begin
                              n =n+1;
               if  (fdigit1  +fdigit2> 9) يوجد باقي //
               begin
                    rem1 =(fdigit1  +fdigit2 ) mod  10;
                    rem2 =(fdigit1  +fdigit2 ) div  10;
                    res  =res +( rem1) *powf(10,n );

                    num1 = num1 + rem2 *10 ;


               end;
               else                         لا يوجد باقي//
                         res = res +((fdigit1+fdigit2) *powf(10,n ));

                    num1=num1  /10;
                    num2 =num2  /10;
               end
          else
               begin
                    n =n+1;
                    res = res +fdigit1 *powf(10,n );
                    break;
               end;
     end;
writeln(res );

end
```

**-الضرب**

```
n=0
while( num2 <>0 )
     fdigit2=num2 mod10
     res=res(+fdigit2*num1()*10^n)
     num2=:num2 div 10
     n =n+1;
```

**المسألة** :6

**اكتب برنامج يقوم بحساب باقي القسمة والقسمة بدون عملية القسمة**

**فكرة الحل كما ورد في تمرين سابق نستمر في الطرح العددين من بعضهما والجواب هو عدد مرات الطرح ولكن الحل هنا بشكل عودي**

```
function div (num1,num2:integer:)integer
begin
if num1=num2
        return 1
else if num1<num2
                res=0
        else
                return 1+sum(num1-num2,num2)
end;
```

-طلب إضافي الMod

```
function mod( num1,num2:integer:)integer
var aux,res :integer;
begin
aux  =num1  div  num2;
aux  =aux *num2;
res  =num1-aux;
mod =res
end;
```

**المسألة** :7

**اكتب برنامج يدخل رقمين مثل** a=1234 و b=5678

**فيصبح** a=1537 و b=2648

```
Readln  a ,b
digit_counter =0
While   a<>0  or   b <> 0 do
        First_a_digit  =a mod 10
        First_b_digit   =b mod 10
        If iseven (First_a_digit  // زوجي  )
                New_a  =new_a   +First_a_digit *pow ( 10,digit_counter)
        If iseven (First_b_digit     )
                New_a  =new_a   +First_a_digit *pow (10,digit_counter)
        If not iseven (First_a_digit  )
                New_b  =New_b  +First_b_digit   *pow( 10,digit_counter)
        If not iseven (First_b_digit )
                New_b  =New_b  +First_b_digit   *pow( 10,digit_counter)
       digit_counter    =digit_counter +1 // تنفذ بكل الحالتين
end {while }
```

**المسألة 8**

- **اكتب برنامج تدخل الرقم** 123456 **فيصبح** 615243
**من اجل الحل نقوم بوضع إجرائيات وتوابع مساعدة**
**التابع** numdigit **يعطي عدد خانات الرقم المدخل**
**التابع** getlast **يعطي آخر خانة مع حذفها مثال الدخل** 123 **يعطي** 23

```
function  getlast (var num:integer :)integer
begin
n1=0 {عداد من اجل حذف آخر خانة بالعدد}
while (num div 10 <> 0)
n1=n1+1
last1=num div 10
num=num div 10 end;{ end of the while loop}

num=num mod (10^n)
getlast=last1
end;تابع
--------------------------------
read     num
ndigit =numdigit(num)
n-=2
while (num div 10 <>0)
begin
fdigit =num mod 10
ldigit=getlast(num)
                                بنلاحظ تغير بقيمة Numلانو هو متحول دخل وخرج
res=res*10( n+2(+(  fdigit*10+ldigit)
num=num div 10
end;
if num div10 =0
ldigit=fdigit=num
res=res*10^(n+2)+(fdigit*10+ldigit)
if ndigit mod2<>0 then
res=res div 10
writeln   res
```

**المسألة 9:**

**اكتب برنامج تدخل الرقم 21643 فيصبح 26413 إي تجعل الأرقام الفردية بجهة والزوجية بجهة أخرى**

```
Read   num
ndigit=getdigit(num)
ndigit1=ndigit
i=1; res=0
while   num div 10 <>0  do
digit =num div 10 ndigit-1
if digit mod2 =0
res=res+digit*10 ndigit1
ndigit1=ndigit
ndigit=ndigit-1
num=num mod 10 ndigit
else
    A[i ]= digit
ndigit=ndigit-1
i++
num=num mod 10 ndigit
end; {end of while}
```

بس تخلص الحلقة بنضيف الأعداد الفردية للناتج

```
for k=1 to i do
res=res+A[i] * 10 i-1
Writeln  res
```

**المسألة 10:**

- **اكتب برنامج يدخل رقم ويقوم بتحويل أكبر رقم مع أصغر رقم**

12093 **يصبح** 12903

**نضع هذا الرقم ضمن** string **ويمكن التعامل معه بكل سهولة اما الطريقة الأصعب هي ان نقوم باخد الرقم خانة خانة والمقارنة بين اكبر قيمة سابقة لل** max **واصغر قيمة سابقة** min **ونحتفظ بعداد للخانات من اجل القيام بالتبديل المناسب**

نرك الحل للطالب كوظيفة :)

**المسألة 11:**

**اكتب برنامج يدخل عدد ويقوم بحذف الأعداد المتكررة** 122335 **يصبح** 1235

```
var  num,n,m=0,f_digit,res=0 ,temp,temp2=0 : long ;

readln(num);
n=0;
while (  num  <> 0 ) do
begin

    f_digit =num mod  10;
    num =num div  10;
    temp=num;
    while (temp <> 0 ) do
    begin
        if(temp mod  10 <>  f_digit) then
        begin
        temp2 =temp2 +(temp mod  10)*powf(10,m);
        m = m+1;
        end;

        temp =temp div 10;

    end
    if(temp =0 ) then
    begin
        res  =res +f_digit *powf(10,n);
        n= n +1;
    end;
    num =temp2;
    temp2=0;
    m=0;

end;
writeln(res);
```

اعتمدنا على وجود تابع powf والذي يعطي 10 مرفوعا للقوة n وذلك وفق برنامجنا السابق

**المسألة** 12:

**اكتب برنامج يدخل عدد من المستخدم ويظهر لنا ناتج جمع الأرقام الأولية والفردية والزوجية من هذا العدد**

**ملاحظة سنقوم باستخدام التابع** primary **في تمرين سابق لمعرفة العدد هل هو اولي ام لا**

```
Readln  n
while n div 10 <>0 do
begin
num=:n mod 10
primary(num)
if primary=true then
      prim=:prim+1;
else
if  num mod 2 =0 then            زوجي
        odd=:odd+1
else
       even=:even+1
end;
```

**الفصل7 مسائل مقرر البرمجة1 العامة**

**المسألة 1: الأعداد الصديقة :**

**يقال عن عددين m,n إنهما عددان صديقان إذا كان مجموع جميع قواسم العدد n=m ومجموع قواسم العدد n يساوي m.والعددان 220 و284 مثال على ذلك فهما عددان صديقان والمطلوب كتابة خوارزمية تقوم بإيجاد الاعداد الصديقة بين 1 و2000 ثم ترجمها إلى لغة باسكال**

```
CONST MAXN=3000;
TYPE DIVIZER=ARRAY[1..MAXN]OF INTEGER;
var                              m,n,i,j,s1,s2:integer;
                                 C:CHAR;
                                A:DIVIZER ;
procedure menu(var c:char);
 begin
          repeat
              WRITELN(,          THE MENU     ')
              writeln('       (C)HECK TOW NUMBER');
              writeln('      (F)RIENDS NUMBER BETWEEN )
             WRITELN(            (E)XIT       ')
             READLN(C);
          UNTIL C IN['C','c','F','f','E','e']
 END;

procedure check(n,m:integer);
var sn,sm,i:integer;
begin
      writeln('INPUT THE FIRST NUMBER');
      READLN(N);
      writeln('INPUT THE SECOND NUMBER');
      READLN(M);
      sn=:0;
      sm=:0;
      for i=:1 to n-1 do
         begin
            if(n mod i =0)then
                sn=:sn+i;
          end;
       for i=:1 to m-1 do
  begin
          if(m mod i =0)then
              sm=:sm+i;
  end;
     if(sm=n)and(sn=m) then
    BEGIN
writeln('AFTER THE CHECKING THE NUMBER  ',N,' AND THE NUMMBER ',M,' ARE FRIENDS');
    END
```

```
    ELSE
        BEGIN
          writeln('AFTER THE CHECKING ....THE NUMBER ',N,' AND THE NUMMBER ',M,' ARE NOT FRIENDS');
        END;
end;

PROCEDURE FRIENDS_BETWEEN(N,M:INTEGER;VAR A:DIVIZER);
VAR i,j,K:integer;

begin
  writeln('INPUT THE FIRST NUMBER ........THE SMALLEST ');
  readln(n);
  writeln('INPUT THE SECOND NUMBER.........THE LARGEST ');
  readln(m);

  FOR I=:1 TO M  DO
     A[I] =0;
  for K=:n to m do
   begin
     FOR J=:1 TO K-1 DO
      BEGIN
         IF(K MOD J  =0)THEN
               A[K] =A[K]+J;
      END;
     FOR I=:N TO M DO
         BEGIN
           IF(A[K] =I)AND (A[I]=K)THEN
   WRITELN('THE NUMBER   ',K,'   AND THE NUMBER   ',I,'   ARE FRIENDS');
          END;
   END;
end;

BEGIN                 {main program }
MENU(C);

WHILE(C<>'E')AND(C<>'e')DO
 BEGIN
  CASE C OF
  'C','c':CHECK(N,M);
  'F','f':FRIENDS_BETWEEN(N,M,A);
   END;
  MENU(C);
  END;
END.
```

**المسألة 2**

**:يعرف العدد الكامل بأنه العدد الذي تساوي قيمته مجموع قيم قواسمه ما عدا العدد نفسه واول عدد كامل هو العدد 6 إذ انه يحقق العلاقة 6=1+2+3 حيث ان 1و2و3 هي قواسم العدد 6**
**والمطلوب كتابة خوارزمية تقوم بإيجاد الأعداد الكاملة حتى ال200 ثم ترجمتها إلى لغة باسكال**

```
const maxn =500;
var
   n,m,i,k:integer;
   a:array [1..maxn]of integer;
begin
  writeln('INPUT THE FIRST NUMBER ........THE SMALLEST ');
  readln(n);
  writeln('INPUT THE SECOND NUMBER........THE LARGEST');
   READLN(m);
  for k=:1 to maxn do
   a[k]  =0;
  for k=:n to m do
   begin
    for i =:1 to k-1 do
      begin
        if k mod i=0 then
           a[k] =a[k]+i;
       end;
  if k=a[k] then
    writeln('the number  ',k,'  is perfect');
   end;
  writeln;
  writeln;
  WRITELN('PLEASE PRESS ENTER TO END');
  readln;
end.
```

**المسألة 3 نريد كتابة برنامج لمعرفة الوضع النسبي لقطعتين إذا كانتا متوازيتين أم متقاطعتين وفي حال تقاطعهما نريد معرفة إحداثيات نقطة التقاطع ووضع نقطة التقاطع [ تنتمي إلى القطعتين، تقع على واحدة من القطعتين وحامل القطعة الثانية تقع على حامل القطعتين والمطلوب**

**1-حدد معطيات الدخل ومعطيات الخرج**

**2-حدد العلاقة بين معطيات الدخل ومعطيات الخرج**

**3-اكتب خوارزمية الحل**

**4-رمز الخوارزمية السابقة بلغة باسكال**

**5-اختبر البرنامج على مجموعة من القيم تغطي كافة الاحتمالات الممكنة**

الحل: **1 - معطيات الدخل : - ميل كل من حاملي القطعتين a1, a2**

- مجال كل من القطعتين [x11,x12] للأولى و[x21,x22] للثانية

- الثابت b لكل من القطعتين ( بفرض معادلة الحامل y = ax + b) أي b1,b2

**2- معلومات الخرج : نتيجة تظهر إذا كانت القطعتان متوازيتان أم متقاطعتان وإذا كانتا متقاطعتان إظهار فيما إذا كانت نقطة التقاطع تنتمي للقطعتين أم لواحد منهما ولحامل الأخرى أو إلى حاملي القطعتين**

**3- العلاقة بين معطيات الدخل ومعلومات الخرج :**

- إذا كان الميلان متساويان a1 = a2 تكون القطعتان متوازيتان، وإلا فالقطعتان متقاطعتان.

- تنتمي نقطة التقاطع ( x,y عندما يكون x ) إلى القطعة عندما يكون x ينتمي إلى مجال تلك القطعة وإلا فالنقطة تنتمي لحامل القطعة.

**4- الخوارزمية كما يلي :**

```
begin
  read (a1)                    قراءة ميلي الحاملين
  read (a2)
  if  (a1  =a2  )then          إذا كان الميلان متساويان فالقطعتان متوازيتان
        print "The two segments are parallel"
  else                         وإلا فهما متقاطعتان
        read (b1)              قراءة ثوابت المعادلتين
        read (b2)

        read (x11)             قراءة مجال القطعة الأولى
        read (x12)

        read (x21)             قراءة مجال القطعة الثانية
        read (x22)

        x ( =b2  -b1) / (a1  -a2)   إيجاد نقطة التقاطع بالحل المشترك للمعادلتين
        y  =a1  *x  +b1

        print "Intersection point is  ") :, x , ", ", y , " "

        if  (x > x11 )and  (x < x12 )then      اختبار إذا كانت النقطة تنتمي للقطعة الأولى
                print "Intersection point is on the first segment"
        else
                print "Intersection point is on first-segment's porter"

        if  (x > x21 )and  (x < x22 )then
                print "Intersection point is on the second segment"
        else
                print "Intersection point is on second-segment's porter"
end
```

**المسألة 5-نريد كتابة خوارزمية وترجمتها إلى برنامج باسكال بحيث تقوم بما يلي :**

**1-قراءة المحارف الممقلة لنص لاتيني علما بان النص ينتهي بحرف . نقطة وحساب طول النص**

**2-إستعراض النص واستبدال كل حرف صغير في بداية إية كلمة بالحرف الكبير الموافق [ فقط في بداية الكلمات جميعها ] واستبدال كل علامة إستفهام موجودة بالنص بعلامة تعجب**

```
var
    c:char;
    orde,s,old,i:integer;
begin
     s=:0;
  c=:'d';            i=:1;
  while c<>'.' do  begin
      read(c)
      s=:s+1;
      orde=:ord(c)
  if ((i=1) or (old=32)) and ((orde>=97) and  (orde<=122)) then
     begin
       write(chr(orde-32))
       i=:2;
     end {end if }
  else if  orde=63 then
      write(chr)33       (({change each ? with !}
   else
       write(c)
       old=:orde;
 end;
writeln('sumall =', s)
readln;  readln;
end.
```

**المسألة 7- إكتب برنامج يحسب N !**

```
fact=:1;
read(  n  )
for i=:1 to n do
                    fact=:fact*n;
writeln('The Factor of the number ',n,'is ',fact);
```

**المسألة** 6:

**نريد كتابة خوارزمية وترجمتها إلى برنامج بلغة باسكال لتقوم بما يلي :**

1-**قراءة المحارف الممثلة لنص لاتيني علماً بأن النص ينتهي بحرف** $ **وحساب طول النص**

2-**إظهار عدد كل من الأحرف الكبيرة والصغيرة وعلامات التنقيط الموجودة في النص بالترتيب**

```
var c:char;
    s,orde,capital,small,digit,dots:integer;
begin
   s=:0;
   capital=:0;
   small=:0;
   digit=:0;
   dots=:0;
   c=:'a';
 while c<>'$' do
   begin
          read(c)
          s=:s+1;
         orde=:ord(c)
          case orde of
                 97..122 :small=:small+1;
                 65..90    :capital=:capital+1;
                 48..57  :digit=:digit+1
             else   dots=:dots+1;
          end;     {end case}
   end;{end while}
writeln('length of the text=', s)
writeln('number of capital letter=',capital)
writeln('number of small letter=', small)
writeln('number of dots=' ,dots)
readln;  READLN ;
end.
```

**المسألة 9-إن مجموع مربعات الأعداد الصحية من 1 إلى n يعطى بالعلاقة**

$$1 + 2^2 + 3^2 + 4^2 + \cdots .... + n^2 = n(n+1)(2n+1)/6$$

**والمطلوب كتابة تابع منطقي يأخذ كوسيط شكلي القيمة n و يعطي نتيجة تدل على صحة العلاقة أو عدم صحتها**

```
var  n:integer;
function sum(n:integer):boolean;
var
    i,s:integer;
    s1:real;  b:boolean;
begin
s1=:0;  s=:0;
for i =:1 to n do
  begin
       s=:s+i*i;
  end;
s1=:n*(n+1*((2*n+1)/6;
if s1=s then
   sum=:true;
end; {end function}
begin  {main program}
write('enter the n=');
readln(n)
writeln(sum(n))
readln;
end.
```

**المسألة** :10

**تشارك الدول العربية التالية في لعبة كرة القدم : سوريا الأردن مصر تونس الجزائر والمغرب تحسب النقاط على النحو التالي**

**خسارة صفر نقطة تعادل واحد نقطة ربح** 2 **نقطة**

**وتتم المباريات بحيث يلعب كل فريق مع كل الفرق المشاركة مرة واحدة . تتوضع النتائج في مصفوفة بحيث يحوي عنصر المصفوفة** Result Country1,country2 **على نتيجة مبارة الفريق** country1 **مع الفريق** country2 **وتعتبر نتيجة الفريق مع نفسه صفراً**

**والمطلوب** 1-**تعريف نمط معطيات سلمي يتضمن الفرق المشاركة**

2-**تعريف نمط معطيات مصفوفة لتمثيل النتائج بحيث يكون الدليلن [ للسطر والعمود ] اهما اسمي الفريقين**

3-**كتابة برنامج يتضمن الأنماط السابقة ويقوم بما يلي :**

- **قراءة النتائج أي ملء المصفوفة ويتم ذلك بافتراض ان المدخل يعرف ان عليه إدخال النتائج العمود تلو العمود وفق ترتيب الفرق أعلاه ولاداعي لإظهار رسائل توضيحية لمساعدته**
- **التحقق من الشرط -**
- **نتيجة لعب الفريق** 1 **مع الفريق** 2 + [ **نتيجة لعب الفريق** 2 **مع الفريق** 1 ] =2 **ولهذا الغرض يجب كتابة تابع يعطي نتيجة منطقية صحيحة في حال تحقق الشرط واستدعاء هذا التابع في البرنامج وغظهار رسالة مناسبة للنتيجة التي يعطيها الاستدعاء**
- **إيجاد الفريق الأول والفريق الأخير في الدوري وإظهارهما-**

```
Program football (input,output);

Type
team= ( syria,jordan,egypt,tunisia,algeria,morocco );
matrix =array [syria..morocco,syria..morocco] of integer;
matrix1  =array [1..6,1..6] of integer;

Var
i,j:team;
a :matrix;
b :matrix1;
t,t1,k,l:integer;

Begin    {main }
 for i =:syria to morocco do
  begin
       for j =:syria to morocco do
    begin
     a[i,j] =0;   { butting zeros in the array }
    end;
  end;
```

```
 writeln('                    Welcom to our program')
writeln(' Please be aware while entering the points that:');
write(' 1 -When you read team1*team2 this means that you should enter');
 writeln(' the points which the team1 got when they played with team2');
writeln(' 2 -You will have a chance to change the value case it is wrong ');
writeln(' 3 -If the team won you should enter 2  & if it lost you should enter 0 & if it drew youshould enter 1');
for i =:syria to algeria do
  begin
   for j=:succ(i) to morocco do
        begin
     case i of
          syria :write(' Syria  *');  jordan :write(' Jordan  *');   morocco :write(' Morocco  *');
          egypt :write(' Egypt  *');  tunisia :write(' Tunisia  *');   algeria :write(' Algeria  *');
         end;  { this case and wherever you see one in this program it is just a way to write a direction in the screen}
     case j of
          jordan :write(' Jordan ');   morocco :write(' Morocco ');
          egypt :write(' Egypt ');  tunisia :write(' Tunisia ');   algeria :write(' Algeria ');
          end;
     readln(a[i,j]);
     writeln;
       while (a[i,j]>2) do
      begin
       writeln('Error');
       writeln(' Re-enter the result ');
       a[i,j] =0;
       readln(a[i,j]);
       writeln;
      end;
    end;
  end;
 for i =:jordan to morocco do
  begin
   for j =:syria to algeria do
      begin
        if  j<i  then
      begin
       case i of
        jordan :write(' Jordan  *');   morocco :write)' Morocco  *');
        egypt :write(' Egypt  *');  tunisia :write(' Tunisia  *');  algeria :write(' Algeria  *');
       end;
       case j of
        syria :write(' Syria ');  jordan :write(' Jordan ');
        egypt :write(' Egypt ');  tunisia :write(' Tunisia ');   algeria :write(' Algeria ');
       end;
       readln(a[i,j]);
       writeln;
        while ((a[i,j]+a[j,i])<>2) do   { examine the condition  }
         begin
          writeln('Error');
          writeln(' Re-enter the result ');
          a[i,j]=:0;
          readln(a[i,j]);
          writeln;
         end;
```

```
                end;
end;
             end;
            for j =:syria to morocco do     { counting the points of each team }
             begin
              b[1,1]=:b[1,1]+a[syria,j];
              b[1,2]=:b[1,2] +a[jordan,j];
              b[1,3]=:b[1,3]+a[egypt,j];
              b[1,4]=:b[1,4]+a[tunisia,j];
              b[1,5]=:b[1,5]+a[algeria,j];
              b[1,6]=:b[1,6]+a[morocco,j];
             end;
            b[2,1]=:ord(syria);
            b[2,2]=:ord(jordan);
            b[2,3]=:ord(egypt);
            b[2,4]=:ord(tunisia);
            b[2,5]=:ord(algeria);
            b[2,6]=:ord(morocco);
            for l=:1 to 5  do
             begin
              for k=:l+1 to 6 do       { ordering }
              if (b]1,k[>b]1,l ([then
               begin
                t=:b[1,l];
                t1=:b[2,l];
                b[1,l]=:b[1,k];
                b[2,l]=:b[2,k];
                b[1,k]=:t;
                b[2,k]=:t1;
               end;
             end;
            writeln;
            writeln(' The order of the teams is ');
            writeln;
            for l=:1 to 6 do
begin
              write(l,'_ ' );
              case b]2,l [of
               0 :write(' Syria with');
               1 :write(' Jordan with');
               2 :write(' Egypt with');
               3 :write(' Tunisia with');
               4 :write(' Algeria with');
               5 :write(' Morocco with');
              end;
              writeln(' ',b[1,l],' point');
              writeln;
             end;
           readln;
           end.
```

**المسألة 11 :**

**نريد كتابة برنامج لمعالجة المحارف يقوم بما يلي**

1-**قراءة نص مكتوب على شكل سلسلة من المحارف تنتهي بحرف** $ **وتخزين هذا النص في شعاع يسمى** Tvect

2-**استنتاج نص جديد يحوي جميع محارف النص الاصلي عدا الفراغ وتخزين النتيجة في شعاع مختلف** Tvectwos

3-**التحقق من كون** Tvectwos **متناظراً أي انه يقرا من اليمين ومن اليسار فيعطي نفس النتيجة ولهذا الغرض يطلب كتابة تابع اسمه** palindrome **يأخذ الشعاع المحتوي على النص كوسيط شكلي ويعطي نتيجة منطقية صحيحة أو خاطئة حسب تناظر النص أو عدمه واستدعاء هذا التابع وغظهار النتيجة الموافقة**

```
type
   countries= (syria,leb,jor,egypt,lybia,tunis,maroco);
var
   a:array[countries] of integer;
   i,j:countries;
   max:integer;
function country(i:countries):string;
var s:string;
begin
  case i of
     syria :s=:'SYRIA  ';
     leb   :s=:'LEBANON';
     jor   :s=:'JORDAN ';
     egypt :s=:'EGYPT  ';
     lybia :s=:'LIPYA  ';
     tunis :s=:'TUNSIA ';
     maroco:s=:'MAROCO ';
  end;
country=:s;
end;
begin

  for i =:syria to maroco do
      begin
          WRITELN;
          writeln('PLEASE ENTER THE NATIONAL INCOME OF ',COUNTRY(I));
          WRITELN;
          write(' THE NATIONAL INCOME OF ',country(i) ,'=');
          readln(a[i])
        end;
     max=:a[syria];
  for i =:leb to maroco do
        begin
          if a[i]>max  then
            begin
              max=:a[i];
              j=:i;
            end;
```

```
    end;
writeln(' THE  COUNTRY WITH THE LARGEST NATIONAL INCOME IS ', country(j),'WITH
',max);
WRITELN('PLEASE PRESS ENTER TO END');
readln;
end.
```

**المسألة 12- تعتمد إحدى الطرائق القديمة لتشفير النصوص اللاتينية بغية جعلها غير مفهومة من قبل العدو على الطريقة التالية :**

**نفترض ان جميع الأحرف الأبجدية موضوعة على محيط دائرة واننا نقوم باستبدال كل حرف بالحرف الذي يبعد عنه بمقدار K باتجاه عقارب الساعة وعلى سبيل المثال إذا كانت k=5 وكانت الأحرف A,B,C,D,E,F,G فأن A يستبدل بالحرف F و B يستبدل بالحرف G و Y يستبدل بالحرف D و Z يستبدل بالحرف E**

**اما بالنسبة للأرقام بين 0 و 9 فنتبع القاعدة التالية**

0<>9 , 8<>1 , 4<>5

**اما علامات التنقيط فتبقى كما هي في النص الأصلي**

**ونفترض ان النص يتم تخزينه برمجياً في شعاع والمطلوب**

**1-كتابة إجرائية للتشفير تأخذ كوسيط K والشعاع المحتوي على النص وتعيد شعاع آخر يحتوي النص المشفر**

**2-كتابة إجرائية لفك التشفير تقوم بعكس العملية السابقة**

**3-كتابة برنامج يقوم بقراءة نص وتشفيره وقراءة نص يفترض انه مشفر ويفك تشفيره**

```
Program CryptoWacko(input,output);
Const              n  =5;
Type              Vector  =array [0..n] of char;
Var             RTVect,ETVect  :Vector;
                      BN,i:Integer;
Procedure Encrypt(k:Integer;Vect:Vector;var eVect:vector);
Var                      I  :Integer;
Begin
 For I  =:0 to n Do
 Begin
  If (ord)vect[I]) > =ord('a' )) AND (ord(vect[I]) < =ord('z' )) Then
           eVect[I] =: chr(ord(vect[I]) + k)
  If ord(eVect[I]) > ord('z' ) Then
            eVect[I] =: chr( ord('a' ) - 1  +(ord(evect[I]) – ord('z' ) ))

  If (ord)vect[I]) > =ord('0' )) AND (ord(vect[I])< =ord('9' )) Then
            eVect[I] =: chr(ord('0' ) + 9  –(ord(vect[I]) – ord('0' )))

  If (ord(vect[I]) > =ord('A' )) AND (ord(vect[I]) < =ord('Z' )) Then
            eVect[I] =: chr(ord(vect[I]) + k)
  If (ord(eVect[I])  > ord('Z' )) AND (ord)vect[I]) > =ord('A' )) AND (ord(vect[I]) < =ord('Z' ))  Then
          eVect[I] =: chr( ord('A' )- 1  +(ord(evect[I]) – ord('Z' )))
 End;
End;
```

```
Procedure Decrypt(k:Integer;Vect:Vector;var eVect:vector);
Var
                I  :Integer;
Begin
 For I  =:0 to n Do
 Begin
  If (ord(vect[I]) > =ord('a' )) AND (ord(vect[I]) < =ord('z' )) Then
       eVect[I] =: chr(ord(vect[I]) - k);
  If ord(eVect[I]) < ord('a' ) Then
       eVect[I] =: chr(ord('z' ) + 1  +( ord(evect[I]) - ord('a' ) ));

  If (ord(vect[I]) > =ord('0' )) AND (ord(vect[I]) < =ord('9' )) Then
       eVect[I] =: chr(ord('0' ) + 9  -(ord(vect[I]) - ord('0' )));

  If (ord(vect[I]) > =ord('A' )) AND (ord(vect[I]) < =ord('Z' )) Then
       eVect[I] =: chr(ord(vect[I]) - k);
 If (ord(eVect[I]) < ord('A' )) AND (ord(ect[I]) > =ord('A' )) AND (ord(vect[I]) < =ord('Z' ))  Then
   eVect[I] =: chr(ord('Z' )+ 1  +(ord(evect[I]) - ord('A' )));
End;
End;

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Begin        {main }
 WriteLn;
 WriteLn(Please Enter The Text You Want To Encrypt, The Length Of The Text Is ',n+1,' :' );
 For I  =:0 to n Do
       Read(RTVect[I]);
 WriteLn;
 WriteLn('Enter The Base Number :' );
 ReadLn(BN);
 Encrypt(BN,RTVect,ETVect);
 WriteLn('The Encrypted Text Is :' );
 For I  =:0 to n Do
        Write(ETVect[I]);
 WriteLn;
 ReadLn;
  WriteLn('Please Enter The Text You Want To Decrypt, The Length Of The Text Is ',n+1,' :' );
 For I  =:0 to n Do
         Read(ETVect[I]);
 WriteLn;
 WriteLn('Enter The Base Number :' );
 ReadLn(BN);
 Decrypt(BN,ETVect,RTVect);
 WriteLn('The Decrypted Text Is :' );
 For I  =:0 to n Do
        Write(RTVect[I]);
 WriteLn;
ReadLn;
End.
```

**المسألة** 13-**نعرف متتالية فيبونانشي بالعلاقة التالية :**

f[0] = 0   f[1] =1        f[n] = f[n-1] + f[n-2]

**المطلوب:**

1-**كتابة تابع عودي لحساب قيمة** fn **من اجل قيمة** n **تعطى كدخل**

2-**كتابة تفصيل الطلبات الداخلية التي تم عند طلب التابع من اجل**n=5 **كتابة تابع غير عودي لحساب** fn **ولا يعتمد على وجود بنية شعاع لتخزين قيم**fn

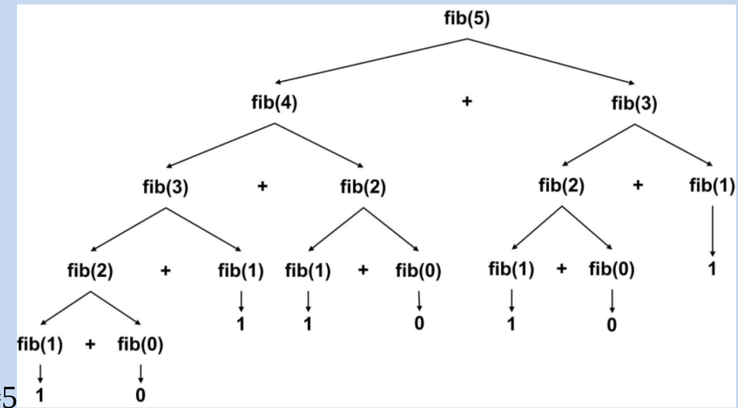
3-**كتابة تابع غير عودي لحساب قيمة** fn **ولا يعتمد على وجود بنية شعاع لتخزين قيم**fn

f[0] = 0   f[1] =1        f[n] = f[n-1[ + f[n-2]

1:

```
Function fib (n:integer)integer;
Readln(n)
If n=0 then
     Fib=:0;
Else if n=1 then
      Fib=:1
Else     fib=:fib(n-1)+fib(n-2)
End;
```

2:

For n=5 

3:

فيبوناتشي بدون عودية

```
Function fib (a:vector; n :integer):integer;
   Begin
If n =0 then
    Fib=:0
Else
    If n=1 then
Fib =:1;
   Else begin
If A[i]=0 then
   A[i]=:fib(n-1)+fib(n-2)
   Fib=:a[i];
End;
```

**المسألة 15-يعرف تابع أكرمان بالعلاقات التالية**

A [0,n] = n+1 ; □

**في حالة n موجبة**

A[m,0] = A[m-1 ,1 ] □

**في حالة m موجبة**

A[m,n] = A[m-1 , A[m,n-1]] □

**في حالة m,n موجبين**

**المطلوب 2-كتابة تابع عودي يسمح بحساب قيمة تابع أكرمان في حالة وسيطين m,n**

**2-تطبيق استدعاء التابع في حالة m=2 و n=1**

```
Function Ack( m,n :integer ) :  integer
Begin
If m =0 then
          Ack  =n+1;
Else if n=0 then
          Ack =A(m-1, 1)
Else
Ack =Ack(m-1,Ack(m,n-1))
End.
```

**المسألة 16 :**

**نريد البحث عن عدد صحيح I بين عناصر قائمة من الأعداد الصحيحة المرتبة بشكل تصاعدي والتي يبلغ عددها n وسنتبع الطريقة التالية في البحث**

**نقارن العدد I بالعدد M الموجود في منتصف القائمة فإذا كان I=M نتوقف عن البحث وإلا نتابع البحث في النصف العلوي من القائمة الأصلية باعتبارها قائمة جديدة وذلك في حال كون I<m أو أو في النصف السفلي من القائمة الأصلية باعتبارها قائمة جديدة وذلك في حال كون I>m ثم نكرر العملية على القائمة الجديدة وهكذا**

**والمطلوب**

1- **كتابة تابع عودي للبحث يأخذ الوسطاء التالية العنصر القائمة ممثلة كشعاع ودليل البداية h ودليل النهاية b ويعيد الدليل الذي وجد فيه العنصر**

2- **كتابة محتوى التابع السابق بخوارزمية غير عودية**

3- **كتابة برنامج يستعمل التابع المعرف إعلاه العودي أو غير إعودي يقوم بقراءة عناصر القائمة المرتبة والعنصر المراد البحث عنه ويستدعي التابع للبحث عنه وإظهار النتيجة**

```
Function Find(A:vertex ; h,b :integer,x:integer ) : integer ;
var     mid :integer;
begin
        mid =trunc (h+b / 2);
    if(b-h =1 ) then
        if A[mid] =x then
           find= mid;
       else
           if A[mid+1]= x then
                 find = mid +1;
           else
              find=0;
      else if A[mid]=k then
                 find = mid;
             else if A[mid]>k then
                  find=find(A,h,b,x);
                   else   find = find(A,mid,b,x);
end;
2:
Function Find (m:vertex ,h,b,k:integer): integer;
var
 w,q:integer;
t:Boolean;
```

```
begin
  t = false;
  repeat
     w=trunk(h+b)/2
      if (b-h =1)then
            if ( m[w]=k)then
                begin
                    q=w;
                    t= true;
                end;
           else  if m[w+1]=k then
                begin
                  q=w+1;
                   t= true;
                end;
                else  q=0;
        else if m[w]=k then
                begin
                     q=w;
                    t=true
              end;
        else if m[w]>k then
             b=w;
              else h=w;
until t or q=0
find =q;
end;
3:

begin  {main}
write('n')
readln(n)
for i =1 to n do
   begin
     write('A[,i,'] = ');
      read(a[i]);
   end;
write('enter  da number wich u search for );
Readln(v);
writeln(find(a,1,n,v);
writeln(find(a,1,n,v);
end.
```

**المسألة 17:**

**نريد قراءة قيمة الدخل الوطني لعدد من الدول وهي سوريا لبنان الأراردن مصر ليبيا تونس المغرب وإيجاد اعلى قيمة بين هذه القيم والمطلوب**

**1-تعريف النمط السلمي المناسب**

**2-تعريف الشعاع الذي سيحوي قيم الدخل الوطني للدول المذكورة**

**3-كتابة برنامج بلغة باسكال يتعامل مع البنى المعرفة في 1و2 ويقوم**

**ب – ملء شعاع الدخل الوطني وفق القيم المدخلة من قبل المستثمر**

- **إيجاد الدولة التي لها اكبر دخل قومي وإظهار رسالة بذلك**

```
Type
    1*Country(sy,leb,jor,egpyt,liybia,tun,mor);
    Var
     income:longint (it's just like integer but with longer range );
    2 *A:array [sy ..mor]of longint;
    3:Begin {main}
    For i=:sy to mor do
    A    :Readln(A[i])
    Max=:A[1]
    B    :For i=:sy to mor do
    If A[i]> max then
        A[i]=:max;
    Else max=:max;
    Writeln('The highest income is ',max )
    Writeln(The country is ord(i)  '(;)  or we can just make case I of and try this one )
    (the ord(i) is very useful tools to know the country )
    End;
```

**المسألة 18 : نريد حساب الجذر التكعيبي لعدد ما باستخدام طريقة نيبوتن التالية**

$X_{n+1} = X_n + 1/3 * (a/X_n^2 - X_n)$

**نتوقف عن الدوران عندما تصبح**

$X_{n+1} - X_n / X_n < eps = 10^{-5}$

**اكتب خوارزمية الحل والبرنامج بلغة باسكال**

```
X :array [1..100] of integer;
Write('Enter the number to count it's cube root ');
Read(a);
X[1]=:1
i=:1;
X[i+1]=:x[i]+(1/3)(a/(x[i])2)-x[i]))
While( (X[i+1] – x[i] /( x[i])>0.00001  do
Begin
i=:i+1;
End;
Sqroot =:(X[i]);
program wasxe;
var i,a:integer;
ss,sq,sqroot:real;
x:array [1..50] of real;

begin
Write('Enter the number to count it's cube root ');
Read(a);
X[1]=:1;
i=:1;
sq=:x[i]*x[i];
ss=:(a/(sq)-(x[i]))/3;
X[i+1]=:x[i]+ss;
While( (X[i+1] - x[i] /( x[i])<0.00001  do
i=:i+1;
Sqroot =:x[i];
writeln('The cube root is',sqroot:10:4);
readln;
readln;

end.
```

**المسألة 19: يسير قطار على خطة سكة حديدية ماراً بالمدن التالية دمشق حمص حماة حلب الرقة دير الزور**

**نريد بناء شعاع معرف على هذه المدن بحيث تكون قيمة كل مركبة هي الساعة التي يمر فيها القطار بهذه المدينة من 0 إلى 24 والمطلوب**

**1-تعريف النمط السلمي اللازم**

**2-تعريف الشعاع الممثل لمواعيد مرور القطار في المدن المذكورة إعلاه**

**3-كتابة برنامج بلغة باسكال يتعامل مع البنى المعرفة في 1و2 ويقوم:**

**ا. ملء المواعيد حسب القيم المدخلة من قِبل المستثمر**

**بطلب من المستثمر إدخال المدينة التي هو فيها والساعة التي يرغب بالسفر فيا من هذه المدينة ويعطيه رسالة تبين كم عليه ان ينتظر أو ربما إن كان عليه الانتظار حتى الغد لان موعد مرور القطار بالمدينة يسبق الساعة التي يردي ان يسافر فيها**

```
program train;
const  nx=6;
type  city = (dam,hom,ham,ale,rak,dir)
var    r:array[dam..dir] of integer;
  k:city ;
 c:char ; t:integer;

proceudre print (o:city);
begin
    case o of
        dam:   write(‚dam‘);
        hom: write(‚homes‘);
        ham :  write(‚Hama‘)
        ale: write(‚Aleppo‘);
        rak: write(‚rak‘);
       dir: write(‚dir ‚);
end;

begin
  for k= dam to dir do
begin
      write(‚The time of the train in ‚): print(k); readln(r[k]);
end;
writeln( اختر أي مدينة تريد الذهاب اليها وذلك بكتابة الحرف المكتوب على شكل كبير)
for k=dam to dir do
begin
print(k); write(‘       ‘);
end;
Readln(c);
```

```
case c of
   'd','D':k:=dam ;
  'h','H': k:=hom;
   'a','A':k:=Ham;
    'l','L': k:= ale;
    'r','R': k:=Rak;
     'i','I': k:=dir;
else   c:=q;
end;
if c<> 'q' then
begin
     write('The Time now is : ');
     read(t);
     if r[k]<t then
              write( 'عذراً لكنك قد تأخرت لقد غادر القطار عند الساعة',r[k])
   else if r[k]=t then
            write( اسرع فالقطار على وشك المغادرة)
else   write ( 'عليك ان تنتظر',r[k]-t);
end;
end;
```

**المسالة 20 لتمثيل أي كثير حدود في الحاسب نحتاج إلى معرفة درجته n وهي عدد صحيح وقيم الأمثال** an.a0,a1,a2,... **وهي إعداد حقيقة والمطلوب**

1- **كتابة تابع يقوم بحساب القوة n لعدد ما X أي** X^n **حيث** X **عدد حقيقي** n **عدد صحيح**

2- **نريد حساب قيمة كثير الحدود عند نقطة ما** x0 **باستخدام طريقة هورنر والمطلوب كتابة تابع ياخذ الوسطاء التالية**

**كثير احدود وفق التمثيل الذي إخترنع له**

**والنقطة** x0 **ويعيد قيمة كثير الحدود عند** x0 **باستخدام العلاقات التالية**

Yn=An

Yn-1 =An-1 +Xyn

.......

Y0 =a0+xyi=P[x]

3- **لنأخذ حالة كثيرات الحدود من الدرجة الثانية نريد حل المعادلة** p[x]=0 **إكتب تابعاً لحساب مميز المعادلة**

4- **إكتب برنامج يقرا كثير حدود** p[x] **ونقطة** X0 **ويستدعي التابع المكتوب في** 2 **لحساب قيمة كثير الحدود وإظهار الناتج بمقارنته مع القيمة الناتيجة من حساب قيمة كثير الحدود باستخدام التابع المكتوب في** 1 **وإذا كان التابع المكتوب في** 1. **وإذا كان التابع من الدرجة الثانية يطلب حل المعادلة** p[x]=0 **باستخدام التابع المكتوب في** 3 **وإظهار جذر أو جذري المعادلة**

```
Program  Poly;
Type             mat =array [0..10 ]of real;

Var                  n,i:integer;
 P:mat     x,r,x1,x2:real

Function h(n:integer,q:mat,x:real):real
Var  s:real;
Begin
S=0;
For i=n downto 0 do
    S=s*x+q[i];
H:=s;
End;

Function pow (x:real,o:integer):real;
Var m:real;
Begin
   M:=1 ;
For i=1 to o do
M:=m*x;
Pow=m;
End;

Function d (m:mat ;d:real):real;
Begin
D=sqr(m[1]-4*m[2]*m[0];
End;


Begin {main program }
 Write('n= '); readln(n);
For i= n downto 0 do
   Begin
Write('a' ,i,'=  ' ); read(p[i]);
  End;

Write('x= ') ;read(x);
Writeln('F(',x:3:1,'(=',h(n,p,x):3:1);

N=p[0];
For i=1 to n do
R=r+p[i]*pow(x,i);
Writeln('f(',x:3:1,')=',r:3:1);
If n =2 then
          If (p[2]=0 ) and (p[1] =0) then
```

```
              Write('Exacusem');
         Else  if (p[2]=0 )then
                 Write('x=',-p[0]/p[1]:3:2)
               Else
    begin
                   X=d(p);
                If x>=0 then

                   Begin

                         X1:=(-p[1]-sqrt(x)/(2*p[2]));
                         X2:=(-p[1]+sqrt(x)/(2*p[2]);
                         Writeln('x1 =', x1:3:2);
                         Writeln('x2= ',x2:3:2);

                     End;
               Else
                  Begin
                X1= -p[1]/(2*p[2]);
                X2:=abs(sqrt(-x)/(2*p[2]);
               Write('x1=',x1,'+',x2);
              Write('x2=',x1:3:2,'-i',x2:3:2);
                   End;
     End;
End.
```

**المسألة 21 يقال عن مصفوفة مربعة بانها ذات قطر مسيطر أو طاغ إذا كانت القيمة المطلقة لكل عنصر موجود على القطر الرئيسي اكبر من مجموع القيم المطلقة للعناصر لوجوده معه في الصف نفسه أي إنه مهما تكن i يكون لدينا**

|Cin|+……+|Ci3|+|Ci2|+|cii| > |Ci1|

**والمطلوب :**

1-**كتابة تابع يقوم بحساب مجموع القيم المطلقة للعناصر الموجودة في صف ما i ما عدا العنصر** Cii

2-**كتابة تابع منطقي يختبر إذا ماكانت مصفوفة ما معطاة له ذات قطر مسيطر ام لا وذلك باستخدام التابع المكتوب في** 1

**ملاحظة :يعرف التابع الاول في موقعه الصحيح ضمن التابع الثاني**

3-**كتابة برنامج يقرا مصفوفة مربعة ويتحقق من كونها ذات قطر طاغ ام لا**

```
CONST MAX  =5;
TYPE  Oh  =ARRAY [1..MAX,1..MAX] OF INTEGER;
VAR  O:Oh;
      I,J:INTEGER;
      BO:BOOLEAN;
Function Big(O:Oh:(boolean;
     var  i:integer;
  Function SumLine_AbsVal(i:integer:(integer;
     type Qu  =array [1..max]of integer;
     var   h,S:integer;
           Q:Qu;
       Begin
         S=:0;
         for h=:1 to i-1 do
           S=:S+abs(Q[h](;
         for h=:i+1 to max do
           S=:S+abs(Q[h](;
       End;
  Begin
    Bo=:true;   i=:1;
    While(i<=max(and(Bo=true(do
      begin
        if(abs(O[i,i](<>SumLine_AbsVal(i((then
          Big=:false;
        i=:i+1;
      end;
    Big=:Bo;
  End;

            BEGIN
              FOR I=:1 TO MAX DO
                BEGIN
                  FOR J=:1 TO MAX DO
                    BEGIN
                      WRITE('        [',I,',',J,']   ='(;
                      READLN(O[I,J](;
                    END;
                END;
                IF (Big(O=(TRUE (THEN
                  WRITELN(' THE RESULT  :IT IS OK .......'(
                ELSE
                  WRITELN(' THE RESULT  :IT IS NOT .......'(;
                READLN;
            eNd.
```

**المسألة 22 لتكن لدينا قائمة من الأعداد اصحية الموجب والمنتهية بالعدد 0 بحيث يمكن ان يرد العدد نفسه ضمن القائمة عدة مرات .نريد إيجاد العدد ذي التواتر الأكبر في القائمة وإظهاره والطلوب**

**1-كتاة تابع يوجد العدد الذي له التواتر الأكبر ضمن قائمة معطاه للتابع**

**2-كتابة تابع يوجد العدد الأكبر بين عددين**

**3+-كتابة برنامج يقرا قائمتين تنتهيان بالعدد 0 ويطلب التابع في 1 لإيجاد العنصرين ذوي التواتر الاكبر فيهما ثم يطلب التابع في 2 لإيجاد العدد الأكبر بينهما**

**4-لو رغبنا في 1 بالحصور على العدد وتواتره فماذا علينا ان نعدل**

التابع الأول:

```
Type Vector  =array [1..100]of integer;
Var  A:Vector;
      i,MAX,N:integer;


Function TheGest_Freq(A:Vector;var most:integer):integer;
    Type Matrix  =array [1..100,1..2]of integer;
    Var  B:Matrix;
(*B is The Result Freq  array 2D (that B[j,1] is for the element /from A/,
B[j,2] is for Freq*)
         max1( *the max of B*),j:integer;
         bo:boolean;
      Begin
        B[1,1]=:A[1];   B[1,2]=:1;    max1=:1;
        i=:2;
        while (i<=Max)  do
          begin
            bo=:false;   j=:1;
            while (j<=max1)and(bo=false) do
             begin
              if (B[j,1]=A[i]) then
                bo=:true;
              j=:j+1;
             end;
            if bo=true then
              B[j-1,2]=:B[j-1,2]+1  (*j-1 becouse the last step were adding 1*)
            else
              begin
                max1=:max1+1; (=*j *) (*new element well be added to B*)
                B[max1,1]=:A[i];   (*creating*)
                B[max1,2]=:1;      ( *it is first time of A[i]*)
              end;
            i=:i+1;
          end;      (*now creating B /array of A elements and its freq*)
        most=:B[1,2];      ( *most here is as representative of the great valu of
Freq*)
        TheGest_Freq=:B[1,1];
        for j=:1 to max1-1 do    ( *texting B *)
          begin
           if (B[j+1,2]>B[j,2]) then
             begin
                      most=:B[j+1,2];
              TheGest_Freq=:B[j+1,1];
             end;
          end;
    End;
```

```
BEGIN
I=:1;   N-=:1;
WHILE (I<=102)AND(N<>0) DO
BEGIN
WRITE('INSERT:');
READLN(N);
A[I]=:N;
I=:I+1;
END;
MAX=:I-2;
I=:TheGest_Freq(A,N);
WRITELN('     NO :',I,'     &^%%^&    FREQ  =',N);
READLN;
END .
```

فكرة الحل:

بفرض A هي القائمة تبع الأعداد

نقوم بإنشاء مصفوفة ثنائية البعدB (بطول=طول مصفوفة قائمة الأرقام المدخلة و وعرضها = 2.....ليش ؟؟ هلق الخانة الاولى منحط فيا العدد و بالتانية عدد مرات تكرار هل العدد

بداية:

أول عنصرمن A نحطو بأول خانة بـB[1,1] ([ B و منحطـB[1,2] = 1(مكرر لمرة واحدة) max1= 1(و هو العدد الاعظمي تبع عناصر B )

منشغل حلقة على عناصر A بدأ من تاني عنصر فيا

طالما ما خلصت A نفذ :(*الحلقة الرئيسة*)

|منقارن كل عنصرA[i ] مع جميع عناصر B (بدأ من j = 1لعند ال max1 ) و ذلك بواسطة حلقة تكرارية شرط كسرها إما انتهاء A أو الوصول لعنصر فيB[j,1] ([ B مماثل لA[i ]

....

بس نخلص من الحلقة ........منشوف شو السبب يلي أدى إلى الخروج منها

>>الوصول للعنصر المطلوب:(تحققB[j,1 ]=A[i] ) نضيف (+1)لخانة B [j,2]

>>وإلا:(خلصنا ال Bكلها و ما كنا نلاقي a[i])مننشأ خانة جديدة ب B بتحمل قيمةA [i] و قيمة تكرارية لها منحطها = 1 و ما مننسى نضيف (+1) لـmax1

________بس لتخلص الحلقة الرئيسة

منشغل حلقة علىB: مشان نحدد أعظم قيمة تكراريةB[j,2] و منشوف قيمة العنصر الموافقة لها B[j,1]

## الفصل 8 مسائل دورات برمجة 1 في كلية الهندسة المعلوماتية

**دورة** 2005-1-13

**السؤال الأول :**

**اكتب بلغة الخوارزميات خوارزمية تقرا سلسة من حروف اللغة الإنجليزية تمثل جملة تنتهي بالحرف نقطة وتحسب عدد الكلمات في هذه الجملة والطول الوسطي لكلمات الجملة مثال لدى ادخال الجملة التالية**

Knowing that many arab countries have taken steps to formulate relevant policies and implementation strategies to built the information society ,it becomes necessary to built upon all such efforts in promoting an integraded plan of action for a future information society in the arab region.

**يكون ناتج الخوارزمية عدد الكلمات** 45 **عدد حروف اللغة الإنجليزية** 246 **الطول الوسطي لكلمات الجملة** 5.47

```
Var c:char;
Letterc,Wordc,, avr:integer;
Read (c);
p=:0;  {actual letter numbers }
while c<>’.’ Do
begin
    letterc=:letterc+1;
     if c=’ ‘ or c=’!’ or c=’#’ {….etc} then  فواصل بين الكلمة واخرى
           wordc=:wordc+1;
       else p=:p+1;
     Read (c);
End;
writeln(‘The number of words are ‘,wordc);
writeln(‘The number of letters ‘,p);
avr=:letterc/wordc;            ( or p/wordc)
writeln(‘The Average ‘,avr );
```

**السؤال الثاني :**

1-**اكتب بلغة باسكال تابعاً اسمه** Capital **دخله حرف من حروف اللغة الإنجليزية** a..z **وخرجه الحرف المقابل حالة** Capital letters

Capital(‘a’)=’A’ ,    Capital(‘f’)=’F’ …. And so on

2-**استخدم التابع السابق في كتابة إجرائية اسمها** MakeFirstCapital **دخلها مصفوفة من المحارف طولها الاعظمي** 30 **تمثل اسم علم وخرجها نفس المصفوفة بعض وضع الأحرف الاولى من الاسم في حالة حرف كبير مثال إذا كان الدخل** salah al  din ben alaziz **يكون الخرج هو** Salah Al  Din Ben Alaziz

```
1:Function Capital (c:char):Char;
Begin
C=:chr(ord(c)-ord(‘a’)+ord(‘A’));
{for example c=:’a’ then c=:chr(ord(a)-ord(‘a’)+ord(‘A’))=chr(ord(‘A’)=’A’  cos the two functions ord and chr are inversed with each other and in this way we don’t need to know the ord by ASCII codes )}
Capital=:c;
End;

Matrix:array[1..30]of char ;

2:Procedure MakeFirstCapitals (var A:matrix);
Begin
    A[1]=:capital(A[1]);
     I=:2;
     While I< =30 do
           If A[i]=’ ‘ then   {space}
                 A[i+1]=:Capital(A[i+1]);
                  I=:i+2;
          Else
                   {A[i]=:A[i];}
                 I=:i+1;
End;
```

**السؤال الثالث: الجدول السحري هو جدول فيه n*n خانة حيث n عدد فردي، يجري ملء المربعات بالأعداد من 1 إلى n2 بحيث إذا جمعنا الأعداد في سطر معين أو عمود معين أو قطر من القطرين فالنتيجة ستكون متساوية بين الكل التالي حالة n=5**

| | | | | |
|---|---|---|---|---|
| 17 | 24 | 1 | 8 | 15 |
| 23 | 5 | 7 | 14 | 16 |
| 4 | 6 | 13 | 20 | 22 |
| 10 | 12 | 19 | 21 | 3 |
| 11 | 18 | 25 | 2 | 9 |

A: **نبدا بالمربع بالأوسط من السطر الأعلى ونضع فيه القيمة 1**
**نحاول في كل مرة الانتقال إلى العمود التالي والسطر الأعلى لنضع القيمة التالية مثل الحالة E وقد نواجه الحالات التالية**
B: **عندما نتجاوز السطر الأعلى نعود إلى السطر الأول**
C: **عندما نتجاوز العمود الأخير نعود إلى العمود الأول**
D: **عندما ننتقل من مربع i,j إلى مربع i+1,j+1 ونجد ان هذا المربع سبق ان وضعنا فيه قيمة فاننا ننتقل إلى المربع i-1,j**
**اكتب بلغة باسكال إجرائية FillMagic Array تقوم بملءجدو ل من الدرجة n حيث n هو دخل الإجرائية ثم اكتب إجرائية إجرائية اسمها nextCell دخلها الموقع الحالة i,j وخرجها إحداثيات الموع الجديد وناخذ بعين الاعتبار الحالات المختلفة التي تم شرحها**

```
Type Matrix = array[1..100,1..100] of integer;
 Var   I,J,N:integer;         Majec:Matrix;         c:Char;□

   Procedure NwxtCell (var i,j:integer);
      var i1,j1:integer;
        Begin
          for i1:=1 to n do        (*this is for marking the M.A with gaps*)
            begin
              for j1:=1 to n do
                Majec[i1,j1]:=-1;
            end;
          i1:=1;      j1:=(n div 2)+1;
          Majec[i1,j1]:=0;
          while ((i1<>i)or(j1<>j)) do   (*//arrive to the wanted cell !//*)
            begin
              i1:=i1-1;     j1:=j1+1;
              if (i1=0) then
                i1:=n;
              if (j1=n+1) then
                j1:=1;
              if Majec[i1,j1]=0 then
                begin
```

```
            if (i=n) then
              i:=0;   (*come Back...*) (* 0 becouse we'll add 2 not 1*)
            i1:=i1+2; (* ???*)
            j1:=j1-1; (* Becouse we have add 1 *)
            if(j1=0) then
              j:=n;   (* !!!!*)
           end;
         Majec[i1,j1]:=0;                (*this cell has been done*)
       end;
     i1:=i1-1;    j1:=j+1;
     if (i1=0) then
       i1:=n;
     if (j1=n+1) then
       j1:=1;
     if Majec[i1,j1]=0 then
        begin   writeln('lll');
          if (i1=n) then
            i1:=0;   (*come Back...*) (* 0 becouse we'll add 2 not 1*)
          i1:=i1+2; (* ???*)
          j1:=j1-1; (* Becouse we have add 1 *)
          if(j1=0) then
            j1:=n;   (* !!!!*)
        end;
     i:=i1;
     j:=j1;
   End;

Procedure FillMajecArray(var Majec:Matrix);
  var  index:integer;
   Begin
     i:=1;     j:=(n div 2)+1;
     for index:=1 to n*n do
       begin
         Majec[i,j]:=index;
         i:=i-1;    j:=j+1;
         if (i=0) then
           i:=n;
         if (j=n+1) then
          j:=1;
         if Majec[i,j] in [1..index] then
           begin
            if (i=n) then
              i:=0;   (*come Back...*) (* 0 becouse we'll add 2 not 1*)
            i:=i+2; (* ???*)
            j:=j-1; (* Becouse we have add 1 *)
            if(j=0) then
```

```
            j:=n;   (* !!!!*)
          end;
       end;
    End;
BEGIN
  writeln;  writeln;  writeln;
  writeln('                    MAJEC ARRAY !!@#!!');
  writeln('                    ==================');
  writeln;writeln;readln;
  repeat
    write('   Before Begining .... Enter The Max : ');
    readln(n);
    if (n>100) Then
     writeln('    ........ its too much !!  Press Enter To Re_Type the Max ');
    if (n<1) then
     writeln('    ........ its too small !!  Press Enter To Re_Type the Max ');
    readln;
  until n in [1..100];
  writeln('   Choose One Of The Following : ');
  writeln;
  writeln('        1 = What Is The Following Cell.');
  writeln;
  writeln('        2 = Fill The Majec Array.');
  writeln;
  writeln('        ............................');
  write('     What do You Want To do ? ');
  readln(c);
  if (c='1')then
    begin
     write(' >>  Type i : ');
     readln(i);
     write(' >>  Type j : ');
     readln(j);
     if(i<1)or(i>100)or(j>100)or(j<1)then
      writeln('       Logical Error !! cannot continue...')
     else
       begin
        NwxtCell (i,j);
        writeln('   The New [',i,',',j,']');
       end;
     readln;
    end
  else if  (c='2')then
    begin
      FillMajecArray(Majec);
      for i:=1 to n do
```

```
        begin
         writeln;
         for j:=1 to n do
         write(Majec[i,j]:6);
         end;
           readln;
      end
    else
      writeln('        General  Error !! cannot continue...');
   End.
```

**الدورة** 2004-1-3

**السؤال الأول :**

**اكتب بلغة الخوارزميات خوارزمية تقرا عددين صحيحن موجبين  يتالف كل منهما من**6 **خانات عشرية ويوجد عدد الخانات العشرية المتساوية في هذين العددين مثال اذا عدد الخانات المتساوية بين العددين**511345 **و** 612368 **هو** 2

2-**بين كيف يجب تعديل الخوارزمية السابقة لتتعامل مع أي عددين صحيحين موجبين مهما يكن عدد خانات الأول والثاني**

الطلب 2:

```
j:=1; freq:=0;
for i:=1 to 6 do{or we can use 2 whiles }
    while j=< 6 do
        If A[i]=B[j] then
                j:=7 {or we can put a Boolean }
                freq:=freq+1
        else
              j:=j+1
```

الطلب :2

تعتمد تلك الطريقة على  string حيث نقرأه ثم نعمل إجرائية checkn للتأكد من ان الرقم السترينغ المدخل هو رقم ثم نفحصه بحيث  لا يهمنا بعد ثابت

```
Function checkn(A:string):Boolean;
begin
For i:=1 to length a do
If a[i]>'0' and a[i]<'9' then
       Checkn:=true;
Else
      checkn:=false;
End;
```

```
Function read (a,b:string):integer;
Freq:=0;
Checkn(a);
If check(a)=true then begin
            Check( b)
 If check(b) true then
      For i:=1 to length(a) do
         While( j<length b) and (g<> true) do
              Begin
                 I f A[i]=B[i] then
                      G=true
                     Freq:=freq+1;
                End;
                                End;
```

[ السؤال الثالث والرابع تم حلهم ضمن المسائل الخارجية ]

**السؤال الأول (7 علامة):**

اكتب برنامجاً بلغة شبه الرماز يقوم بقراءة قيمتين لمتحولين صحيحين، يبدل بين هاتين القيمتين، ثم يقوم بطباعة المتحولين بعد التبديل. **لا يمكنك استخدام أي متحول وسيط.** مثال: بعد قراءة **A=4** ، **B=2** يجب طباعة المتحــولين على النحو التالي: **A=2 B=4**.

**السؤال الثاني (7 علامة):**

عميل في بنك رصيده **balance** يتقاضى فائدة سنوية مئوية **rate**. اكتب برنامجاً بلغة شبه الرماز يقــوم بقراءة المتحولات السابقة، ثم يقوم بحساب وطباعة عدد السنوات التي يتضاعف فيها رصيد العميل (لا يقوم العميــل بسحب أي مبلغ).

**السؤال الثالث (25 علامة):**

اكتب برنامجاً بلغة باسكال يقوم بقراءة عددين صحيحين ثم استدع فيه التابعين التاليين بعد كتابتهما:

- تابعاً تراجعياً (عودياً) **Sum** بلغة باسكال يقوم بجمع العددين بطريقة تراجعية وطباعة الناتج.
- تابعاً **GCD** يوجد القاسم المشترك الأعظم **للعددين معاً.** مثال: القاسم المشترك الأعظم للعدد **150** والعدد **90** هو العدد **30**. عودياً

**السؤال الرابع (15 علامة):**

لدينا مصفوفة اسمها **Vec** (نسق: مصفوفة عدد أسطرها واحد) عدد عناصرها صحيحة عددها ثابت **L**. اكتب برنامجاً بلغة باسكال تقوم فيه باستدعاء التابعين التاليين بعد كتابتهما:

- تابع **FindElem** تمرر إليه المصفوفة **Vec** ويرد موقع عنصر ضمن النسق في حال وجوده، أو يــرد القيمة الخاصة (1-) في حال عدم وجوده.
- تابعاً **RLE** تمرر إليه المصفوفة **Vec** ويرد مصفوفة (نســق) اســمها **Comp** ويــرد عــدد عناصــرها **Count**. ويعمل التابع على النحو التالي:
  - في حال كون عدد الأرقام الصحيحة المتماثلة والمتتالية في المصفوفة **Vec** أقل أو يساوي أثنين يقوم التابع **RLE** بنسخ تلك العناصر من **Vec** إلى **Comp**.
  - في حال كون عدد الأرقام الصحيحة المتماثلة والمتتالية أكبر من أثنين يقوم التابع **RLE** بنسخ عــدد هذه العناصر والعنصر المكرر.

**مثال:**

Vec

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 2 | 2 | 3 | 3 | 3 | 3 | 0 | 2 | 2 | 2 | 1 | 1 |

Comp

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 3 | 0 | 2 | 2 | 4 | 3 | 0 | 3 | 2 | 1 | 1 |

السؤال الخامس (16 علامة):

أكتب برنامجاً بلغة باسكال يقوم بقراءة نص من لوحة الملامس يحوي مجموعة من المحارف اللاتينية والأرقام (200 محرف). ينتهي النص بالمحرف "!"، ثم قم باستدعاء التابعين التاليين بعد كتابتهما:

- تشفير النص بطريقة الإزاحة وحفظه ضمن نسق (مصفوفة عدد أسطرها واحد):
  - الأحرف اللاتينية: إزاحة كل حرف مقروء بمقدار عدد ثابت موجب و صحيح **K**.
  - الأرقام: (0 يرمز بـــ 9) و (1 يرمز بـــ 8) و (2 يرمز بـــ 7) وهكذا...
- فك تشفير النص وطباعته على الشاشة. ملاحظة : يمكن استخدام التوابع المسبقة التعريف.

## السؤال الخامس

```
program  Enc;
const
  maxn=200;
type
    ve=array[1..maxn] of char;
var
     a1,a2,a3:ve;   n,i,j,k,s:integer;
procedure  tash(a1:ve; k:integer; var a2:ve);
var s:integer;
begin                  { tashfeer procedure}
  for i:= 1 to n do
    begin
         s:=ord(a1[i]);

         if ((s>=65) and(s<=90-k)) or((s>=97) and(s<=122-k))then
           begin
            s:=s+k;{A..Z-k or a..z-k}
            a2[i]:=chr(s);
           end
         else if(s>90-k)and(s<=90)then
          begin
            s:= 65+(k-(90-s)-1);
            a2[i]:=chr(s);
          end
         else if(s>122-k)and(s<=122)then
          begin
           s:= 97+k-(122-s)-1;
           a2[i]:=chr(s);
          end
         else if(s>=48)and(s<=57)then{numbers}
```

```
          a2[i]:=chr((57-s)+48)
         else
            a2[i]:=a1[i];
         end;
end;
procedure f_tash(a1:ve; k:integer; var a2:ve);
var z:integer;
begin           { not tashfeer procedure (original text)}
    for i:= 1to n do
       begin
           S:=ord(a1[i]);
           if((s>=65+k)and(s<=90))or((s>=97+k)and(s<=122))then
            begin
             s:=s-k;
             a2[i]:=chr(s);
            end
           else if(s>=97) and(s<97+k) then
            begin
             s:=122 -(k-(s-97)-1);
             a2[i]:=chr(s);
            end
           else if(s>=65)and(s<65+k)then
            begin
              s:= 90 -(k-(s-65)-1);
              a2[i]:=chr(s);
            end
           else if(s>=48)and(s<=57)then{numbers}
               a2[i]:=chr((57-s)+48)
           else
               a2[i]:=a1[i];
        end;  {end for}
end;    {end procedure}
begin
 write('INPUT THE key.... ');
 readln(k);
 i:=1;
 write('INPUT LONG OF THE TEXT (number of chars) n=');
 readln(n);
 write('INPUT THE TEXT :  ');
 WRITELN;
 for i:= 1 to n do
     read(a1[i]);
 tash(a1,k,a2);
 writeln;
 writeln('---------------------------');
 write(' THE TEXT AFTER TASHFEER : ');
```

```
 WRITELN;
 for i:= 1to n do
    write (a2[i]);
 writeln;
 f_tash(a2,k,a3);
 writeln;
 writeln('----------------------------');
 write('THE TEXT AFTER DELET THE TASHFEER (original text) IS :');
  for i:= 1 to n do
    write(a3[i]);
end.
```

## السؤال الثالث

```
program math;
var
  i,c,d:integer;
  a,b:real;
function sum(a:real; b:real):real;
begin
 if (b=0) then
  sum:=a
 else if(b >0)then
    sum:=1+sum(a,b-1)
 else
    sum:=-1 +sum(a,b+1);
end;
function GCD(var c,d:integer):INTEGER;
var max,min:integer;
    f:boolean;
begin
 if(c>d)then
  begin
   max:=c;
   min:=d;
  end
 else
  begin
   max:=d;
   min:=c;
  end;
 if(max mod min = 0)then
   gcd:=min
 else
  f:=false;
   begin
    i:= (min div 2);
```

```
     while(i>0)and(f=false) do
      begin
        if(max mod i =0) and(min mod i =0)then
          begin
           gcd:=i;
           f:=true;
          end;
       i:=i-1;
      end;
   end;
end;
begin
 writeln('FIRST INPUT TWO NUMBER TO FIND THE SUM ');
WRITE('INPUT THE FIRST NUMBER.....');
 READLN(A);
 WRITE('INPUT THE SECOND NUMBER... ');
 READLN(B);
WRITELN('THE SUM =',SUM(A,B):5:2);
WRITELN('INPUT TWO NUMBER TO FIND THE GREAT COMMON DIVIDER ');
WRITE('INPUT THE FIRST NUMBER...');
 READLN(C);
  WRITE('INPUT THE SECOND NUMBER..');
 READLN(D);
  WRITELN('THE GREAT COMMON DIVIDER  = ',GCD(C,D));
 end.□
```

**السؤال الثاني :**

```
PROGRAM BANK_MAN;
VAR balance,y,n:integer;
    rate:real;
begin
 writeln('INPUT THE FALUE OF BALANCE');
 readln(balance);
  writeln('INPUT THE FALUE OF RATE ');
 readln(rate);
 n:=balanc*2
 y:=0;
 while (balance <= n)do
  begin
   y:=y+1;
   balance:=balance+balance*rate;
  end;
 write('THE NUMBER OF YEAR = ',Y);
end.□
```

## السؤال الرابع

```
program exam_2007_4;
type            vector = array [1..30] of integer;
var               a,b:vector;
                k,c,count,l,z:integer;

function findelem (a:vector;n,x:integer):integer; { to find element in array }
var
  found:boolean;
  i:integer;
begin
  found:=false;
  i:=0;
  while (i<=n)  and (not(found)) do
     if (a[i]=x) then
       begin
       found:=true;
       findelem:=i;
       end
     else
     begin
      i:=i+1;
      findelem:=-1;
      end;
   end;

function rle (vec:vector;n:integer;var comp:vector):integer; { to compress array }
var
  i,j:integer;
begin
  i:=1;j:=1;
  while (i<=n) do
    begin
    if (vec[i]<>vec[i+1]) then
      begin
      comp[j]:=vec[i];
      i:=i+1;
      j:=j+1;
      end
    else if (vec[i+1]<>vec[i+2])then
      begin
      comp[j]:=vec[i];
      comp[j+1]:=vec[i+1];
      i:=i+2;
      j:=j+2;
```

```
    end
   else
    begin
    c:=1;
    while (vec[i]=vec[i+1]) and (i<n) do
     begin
     c:=c+1;
     i:=i+1;
     end;
    comp[j]:=c;
    comp[j+1]:=vec[i];
    j:=j+2;
    i:=i+1;
    end;
    end;
    rle:=j-1;
    end;

begin
writeln('please enter the length of the array ');
 read(l);
 writeln;
 writeln('please insert the elements of the array');
 for k:=1 to l do
   read(a[k]);
writeln('please enter the element that you want to search about it ' );
 read(z);
 writeln;
if (findelem(a,l,z)=-1) then
 writeln('this element is not existed in the array')
else
 writeln('this element is existed in the array in location ' ,findelem(a,l,z) );
writeln('the array after compressing ');
 count:=rle(a,l,b);
   for k:=1 to count do
     write ( b[k],' ');
 end.
```

**السؤال الأول تمت الإجابة عنه في قسم المسائل الخارجية**

**الدورة** 2007-2008

**السؤال الأول**

**اكتب برنامج بلغة باسكال تعرف فيه نسق اسمه** vec2d **عدد عناصره الصحيحة** $n^2$ **ويمثل مصفوفة مربعة كل** n **عنصر في النسق هو عبارة عن سطر في المصفوفة المربعة ثم اكتب الإجرائيات التالية:**

1- **الإجرائية** ReadMat **وتقوم بقراءة عناصر المصفوفة المربعة من لوحة الملامس**

2- **الإجرائية** PrintMat **تقوم بطباعة عناصر المصفوفة بحيث تتم طباعة كل سطر في المصفوفة على سطر مستقل**

3- **الإجرائية** Sumdiag **تقوم بجمع عناصر القطر الرئيسي في المصفوفة ويرد المجموع**

4- **الإجرائية** summat **وتقوم بجمع عناصر مصفوفتين ويرد الناتج في مصفوفة ثالثة نسق ثالث**

5- **الإجرائية** PrintMaxAvg **تقوم بطباعة رقم السطر ورقم العمود لكل عنصر في المصفوفة قيمته اكبر من وسطي عناصر المصفوفة**

6- **الإجرائية** ProdMat **تقوم بضرب عناصر مصفوفتين ويرد الناتج في مصفوفة ثالثة نسق ثالث**

7- **الإجرائية** TransposeMat **وتقوم بإيجاد منقول المصفوفة ويردها من خلال مصفوفة ثالثة**

8- **الإجرائية** SortMat **تقوم بفرز عناصر كل سطر في المصفوفة على حدة تصاعدياً ويرد المصفوفة المفروزة**

9- **الإجرائية** CompressMat **وتقوم بضغط عناصر عناصر كل سطر في المصفوفة على حدة على النحو التالي قيمة كل عنصر في السطر بعد الضغط ما عدا العنصر الأول الذي يبقى على حاله هي الفارق بين قيمته السابقة وقيمة لعنصر التالي له ويرد النسق المضغوط**

**مثال من اجل نسق فيه** n=6 **من اجل سطر ما**

**عناصره قبل الضغط هي على النحو التالي**100101105110110120

**عناصره بعدالضغط هي على النحو التالي**100145010

10- **الإجرائية** DecompressMat **وتقوم بفك ضغط عناصر كل سطر على حدة بطريقة عاكسة لآلية الضغط السابقة**

**ثم عرف** mat1,mat2,mat3 **من النمط** Vec2d **وثم باستدعاء الاجرائيات على الترتيب التالي :**

**قراءة نسقين** mat1,mat2

**طباعة مجموع عناصر القطر الرئيسي للنسق** mat1

**جمع عناصر النسقين** mat1,2 **ويرد الناتج في** mat3 **ويطبعه**

**استدعاء الإجرائية** PrintMaxAvg **من اجل النسق** mat3

**ضرب عناصر النسق** mat1**و** mat2 **ويرد الناتج في** mat3 **ويطبعه**

**إيجاد منقول النسق** mat1 **ويرد الناتج في** mat3 **ويطبعه**

**فرز النسق** mat1 **ويرد الناتج في** mat3 **ويطبعه**

**ضغط النسق** mat1 **ويرد الناتج في** mat1 **ويطبعه**

**فك الضغط للنسق** mat1 **ويرد الناتج في** mat1 **ويطبعه**

```
program exam;
type
vec2d=array[1..100] of integer;
var
                    mat1,mat2,mat3:vec2d;
                       m,s:integer;

procedure readmat (var vec:vec2d; n:integer);
var
i:integer;
begin
for i:=1 to n*n do
   read(vec[i]);
end;

procedure printmat(vec:vec2d;  n:integer);
var
i,j:integer;
begin
for j:=0 to n-1 do
   begin
   for i:=(j*n)+1 to n*(j+1) do
      write(vec[i],' ');
      writeln;
   end;
end;
□




procedure sumdiag(vec:vec2d;  n:integer; var sum:integer);
var
i,j:integer;
begin
sum:=0;
for i:=0 to n-1 do
  sum:=sum+vec[i*n+1];
end;

procedure summat(a,b:vec2d;   var c:vec2d; n:integer);
var
i:integer;
begin
for i:=1 to (n*n) do
```

```
  c[i]:=0;
for i:=1 to (n*n) do
  c[i]:=a[i]+b[i];
end;

procedure printmaxavg ( vec:vec2d;n:integer);
var
  j,k,sum,i:integer;
  avg:real;
begin
  sum:=0;
  for k:=1 to n*n do
    sum:=sum+vec[k];
  avg:=sum/(n*n);
  for k:=1 to n*n do
    if (vec[k]>avg) then
       begin
       i:=k div n +1;
       j:=k mod n ;
       if (j=0) then
         begin
         i:=k div n ;
         j:=n;
         end;
       writeln('(',i,',',j,')');
       end;

   end;




procedure prodmat (a,b:vec2d; var c:vec2d ; n:integer);
var
   i,j,k:integer;
begin
for i:=1 to n*n do
   c[i]:=0;
  j:=0;
  k:=1;
  while j<n do
    begin
    while (k<=n*(j+1)) do
      begin
      i:=0;
      while (i<n) do
```

```
        begin
        c[k]:=c[k]+a[n*j+(i+1)]*b[n*i+(k-j*n)];
        i:=i+1;
        end;
      k:=k+1;
      end;
    j:=j+1;
    end;
  end;

procedure transposemat (a:vec2d; var b:vec2d;n:integer);
var
  i,j,k:integer;
begin
  k:=1;
  for j:=1 to n do
    for i:=0 to n-1 do
      begin
      b[k]:=a[i*n+j];
      k:=k+1;
      end;
end;

procedure sortmat (vec:vec2d;var a:vec2d; n:integer);
var
   i,q,k,t,s,e:integer;
begin
    for q:=1 to n do
       for k:=1 to n-1 do
        for i:=(q-1)*n+1 to q*n-k do
          if (vec[i]>vec[i+1]) then
             begin
```

```
            t:=vec[i];
            vec[i]:=vec[i+1];
            vec[i+1]:=t;
            end;
    for i:=1 to n*n do
      a[i]:=vec[i];

end;

procedure compressmat (var vec:vec2d; n:integer);
var
   i,j:integer;
begin
   for j:=n downto 1 do
     for i:=n*j downto (n*j)-n+2 do
       vec[i]:=vec[i]-vec[i-1];
end;

procedure decompressmat (var vec:vec2d ; n:integer);
var
   i,j:integer;
begin
   for j:=0 to n-1 do
     for i:=j*n+2 to (j+1)*n do
       vec[i]:=vec[i]+vec[i-1];
end;






begin{main}
writeln('please enter the length of array');
readln(m);
writeln;
writeln('please enter the contents of the first array');
readmat(mat1,m);
writeln;
writeln('please enter the contents of the second array');
readmat(mat2,m);
writeln;
sumdiag(mat1,m,s);
writeln('the sum of the main diagonal''s elements is ');
writeln(s);
writeln;
```

```
summmat (mat1,mat2,mat3,m);
writeln('the result of sum the two array ( the third array) ' );
printmat(mat3,m);
writeln;
writeln('the places of the elements which bigger than the averge in the third array are ');
printmaxavg(mat3,m);
writeln;
prodmat(mat1,mat2,mat3,m);
writeln('the result of multiplication the two array ( the third array) ' );
printmat(mat3,m);
writeln;
transposemat (mat1,mat3,m);
writeln('the transported of the first array');
printmat (mat3,m);
writeln;
sortmat(mat1,mat3,m);
writeln('the first array after sorting');
printmat(mat3,m);
writeln;
compressmat(mat1,m);
writeln ('the first array after compression');
printmat(mat1,m);
writeln;
decompressmat(mat1,m);
writeln ('the last array after decompression');
printmat(mat1,m);
writeln;
readln;
end.
```